UNCLASSIFIED

AD NUMBER
AD860060
NEW LIMITATION CHANGE
TO **Approved for public release, distribution unlimited**
FROM **Distribution authorized to U.S. Gov't. agencies and their contractors; Critical Technology; JUL 1969. Other requests shall be referred to Army Enineer Topographic Laboratory, Attn: ETL-GS [Geographic Information Systems], Fort Belvoir, VA 22060.**
AUTHORITY
USAETL ltr, 1 Oct 1970

THIS PAGE IS UNCLASSIFIED

AD 860060

SPECIAL INTERIM TECHNICAL REPORT
THE BAYESIAN APPROACH TO IDENTIFICATION OF A REMOTELY SENSED ENVIRONMENT

by
Robert Haralick

CRES Technical Report No. 133-9
July 1969

Sponsored by
ADVANCED RESEARCH PROJECTS AGENCY, DEPARTMENT OF DEFENSE
Work Order No. 1079

STATEMENT #2 UNCLASSIFIED

This document is subject to special export controls and each transmittal to foreign government or foreign nationals may be made only with prior approval of ____________________

Monitored by
U.S. ARMY ENGINEER TOPOGRAPHIC LABORATORIES
GEOGRAPHIC INFORMATION SYSTEMS BRANCH
GEOGRAPHIC SYSTEM DIVISION
FT. BELVOIR, VIRGINIA
attn: ETL.GS.

Contract No. DAAK02-68-C-0089

CRES
THE UNIVERSITY OF KANSAS • CENTER FOR RESEARCH INC
ENGINEERING SCIENCE DIVISION • LAWRENCE, KANSAS

69

AD ______________

SPECIAL INTERIM TECHNICAL REPORT

THE BAYESIAN APPROACH TO IDENTIFICATION OF A REMOTELY SENSED ENVIRONMENT

Robert Haralick

CRES Technical Report No. 133-9

July 1969

Sponsored by
ADVANCED RESEARCH PROJECTS AGENCY, DEPARTMENT OF DEFENSE
Work Order 1079

Monitored by
U.S. ARMY ENGINEER TOPOGRAPHIC LABORATORIES
GEOGRAPHIC INFORMATION SYSTEMS BRANCH
GEOGRAPHIC SYSTEM DIVISION
FT. BELVOIR, VIRGINIA

Contract No. DAAK02-68-C-0089

ABSTRACT

The first part of this paper provides a brief tutorial introduction of the Bayesian Approach to identification of a remotely sensed environment. The second part describes the input data deck setup for the Fortran IV program which has been written to implement this approach. The third part describes file usage and subroutine organization. The fourth part provides a listing of the program with a simple sample data set.

ACKNOWLEDGEMENT

I wish to acknowledge the assistance of Carl Smith who did most of the programming, and the University of Kansas Computation Center for providing some of the necessary computer time.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

PART I

THE BAYESIAN APPROACH TO IDENTIFICATION OF A REMOTELY SENSED ENVIRONMENT

I. THE BAYESIAN APPROACH TO IDENTIFICATION OF A REMOTELY SENSED ENVIRONMENT

Using remote sensors we can make measurements of an environment. The set of measurements made will be called the data set. Our job is to examine the data set in order to identify what the environment is made up of: our problem is how should we do it? In what follows we describe the Bayesian decision approach with a deterministic decision rule.

We assume that distinct boundaries enclose a limited environment, which is made up of small-area patches, one next to the other. The identification of the environment consists of identifying each small-area patch within one category of a given set of categories. We assume that such an identification is sensible and possible.

In order to make any identification we must have knowledge concerning which kind of measurements are typical measurements of the categories we wish to identify. This knowledge is succinctly contained in a classification, which is a mapping, associating with each measurement the category to which it is most typical - given a specific decision criterion. Therefore, if we are to identify measurements in a data set we must have a classification.

How do we obtain a classification? We perform an information gathering experiment. From the population of all environments, we sample one or a few in which it is possible to identify many small-area patches within each category of interest. The proportion of occurrence of each category in the sampled environment(s) does not have to be representative of the average probability of occurrence of each category in the entire environmental population. However, if we have no information regarding the average probability of occurrence of each category in the environmental population, then we would want to choose the sampled environment(s) so that the proportion of occurrence of each category in the sampled environment(s) is an unbiased estimate of the

average probability of occurrence of each category in the environmental population. In either case, the small-area patches within each of the sampled environment(s) do have to be representative of the categories with which they are identified.

With each of our sensors, we measure each small-area patch in the chosen environments. From photo-interpretation or field studies, the environments are examined first hand, and an identification of each small-area patch is made. The sequence of such identifications is called the "ground truth identification" or simply "ground truth". It is from the data set (the sequence of measurements) and the ground truth (the sequence of identifications) that we can find a Bayes classification.

At this point we must introduce some mathematical notation. Let $C = \{c_i\}_{i=1}^{K}$ be the set of K given categories; c_i is the symbol used for the i^{th} category. We suppose, for convenience, that each sensor produces only one number for each measurement it makes of a small-area patch We suppose further, that the j^{th} sensor must produce a number belonging to its range set $L_j = \{l_{j1}, l_{j2}, \ldots, l_{jN_j}\}$. This supposition is fully in accord with reality, since the output of any sensor is always equivalent to a pointer-reading on a dial. Pointer-readings can never be discerned precisely, and are thus discerned approximately to third, or fourth, or, . . . , N^{th} place accuracy.

Measurement space M is the set of all measurements which are possible to make with the set of S sensors. M is conveniently described as the cartesian product of the range sets; $M = L_1 \times L_2 \times \ldots \times L_S$. This is the set of measurements which contain for elements, all the possible numbers produced by sensor one, combined with all the possible numbers produced by sensor two, . . . , combined with all the possible numbers produced by sensor S. For convenience we number the measurements in M: $M = \{m_n\}_{n=1}^{N}$, where N is the total number of elements in measurement space. Finally we must provide a goodness criterion; thus, we introduce a gain function g. $g(c_i, c_j)$ is our economic gain if we identify a measure-

ment as belonging within the i^{th} category when that measurement was made of a small-area patch actually belonging within the j^{th} category.

We have already mentioned that a classification is a mapping or rule which associates with each measurement m_n in M, the category c_i to which it is most typical - according to some decision criterion. Our decision criterion is economic; "most typical to" translates to, "that association by which we, on the average, gain the most economically". Therefore, according to our decision criterion, we can judge each possible classification. That classification which enables us to gain the most, on the average, is the classification which is best; it is that classification which we wish to find.

Let us now examine how the average gain may be calculated. Let f be a classification mapping. f is a function whose domain is the set M, and whose range is the set C; f: $M \to C$. For each element $m_n \in M$ the function associates one and only one category $c_i \in C$. We define the characteristic function h_f for f as follows: for every $m_n \in M$, $c_i \in C$.

$$h_f(c_i, m_n) = \left.\begin{array}{l} 1 \text{ if and only if } f(m_n) = c_i \\ 0 \text{ otherwise} \end{array}\right\} .$$

In other words $h_f(c_i, m_n)$ is 1 if and only if the classification f identifies the measurement m_n as belonging within the category c_i. The average gain A for the classification f is easily seen to be:

$$A(f) = \sum_{i=1}^{K} \sum_{k=1}^{K} \sum_{n=1}^{N} g(c_k, c_i) h_f(c_k, m_n) P(m_n | c_i) P(c_i)$$

where $P(m_n | c_i)$ is the conditional probability that the measurement m_n will be made of a small-area patch given that the patch belongs within category c_i. $P(c_i)$ is the probability that any small-area patch of the

environments in the population belongs within category c_i, and $g(c_k, c_i)$ is the amount gained if a patch which actually belongs within category c_i is identified within category c_k.

Of the four terms in the summation, $g(c_k, c_i)$ is specified as part of the identification goodness criteria, $h_f(c_k, m_n)$ is defined from the classification f, $P(m_n | c_i)$ will be determined from the data gathered in the experiment, and $P(c_i)$ is an additional <u>a priori</u> probability which we will have to specify. Let us now examine in detail how the conditional probabilities are determined from the experimental data.

The data set is a sequence D of R measurements;

$$D = \left\langle m_{r_1}, m_{r_2}, \ldots, m_{r_R} \right\rangle .$$

The ground truth corresponding to sequence D is a sequence T of R not necessarily different category identifications; $T = \left\langle c_{r_1}, c_{r_2}, \ldots, c_{r_R} \right\rangle$. Let # be the counting measure. #(D) is the number of elements in the sequence D; thus, #(D) = R. A sequence is really a function whose domain is the set of integers I. The data set D is then a function which associates with each integer, a measurement; $D: I \rightarrow M$. The ground truth T is also a function and it associates with each integer a category; $T: I \rightarrow C$. D(7), for example, is then just the seventh element in the sequence D; $D(7) = m_{r_7}$. $D^{-1}(m)$ is the set of all integers i for which $D(i) = m$. The statistic $\hat{P}(m_n | c_i)$ estimating $P(m_n | c_i)$ is defined as

$$\left.\begin{aligned} \hat{P}(m_n | c_i) &= \frac{\#(D^{-1}(m_n) \cap T^{-1}(c_i))}{\#(T^{-1}(c_i))} \quad \text{when } \#(T^{-1}(c_i)) \neq 0 \\ &= 0 \text{ otherwise} \end{aligned}\right\} .$$

$\hat{P}(m_n|c_i)$ is the number of integers which are associated with the measurement m_n in the sequence D and with the category c_i in the sequence T, divided by the number of integers associated with the category c_i in sequence T. Stated simply, $P(m_n|c_i)$ is just the number of times the measurement m_n was made of a small-area patch belonging within category c_i, divided by the number of times a small-area patch belonged within the category c_i.

The a priori probabilities $P(c_i)$ can either be estimated from the sampled data set (if this data set is representative of the population) or from our foreknowledge of the population of environments. If we can assume that the few environments we have chosen to sample for our experiment are representative of the population, then

$$\hat{P}(c_i) = \frac{\#(T^{-1}(c_i))}{R}$$

is a reasonable estimate. If we cannot make such an assumption and we believe that a small-area patch is just as likely to belong within one category as within another, then $\hat{P}(c_i) = 1/K$ is a reasonable estimate.

From the estimates $\hat{P}(m_n|c_i)$ and $\hat{P}(c_i)$ we may estimate the average gain $\hat{A}$ for any classification f. As before let h_f be the characteristic function for f.

$$\left.\begin{aligned} h_f(c_k, m_n) &= 1 \text{ if and only if } f(m_n) = c_i \\ &= 0 \text{ otherwise} \end{aligned}\right\} .$$

$$\hat{A}(f) = \sum_{i=1}^{K} \sum_{k=1}^{K} \sum_{n=1}^{N} g(c_k, c_i) h_f(c_k, m_n) \hat{P}(m_n|c_i) \hat{P}(c_i).$$

We seek the Bayes classification f* which maximizes $\hat{A}$. f* is easily defined. For each measurement m_n and for any classification f,

there will be one and only one category c_j such that $h_f(c_j, m_n) = 1$. Consider the amount $\hat{a}(c_j, m_n)$ gained due to the identification of measurement m_n as belonging within category c_j.

$$\hat{a}(c_j, m_n) = \sum_{i=1}^{K} g(c_j, c_i)\hat{P}(m_n | c_i)\hat{P}(c_i)$$

The maximum $\hat{A}(f)$ is certainly achieved if for each measurement m_n, $f(m_n) = c_j$ where c_j maximizes $\hat{a}(c_j, m_n)$. Therefore we just have to compute $\hat{a}(c_j, m_n)$ for $j = 1, 2, \ldots, K$ to determine which category, c_j, maximizes it. Then we define $f^*(m_n) = c_j$.

In this manner we can define how to best identify each measurement which actually occurred in the data sequence D. However, there may be many measurements in measurement space M which did not occur in the data sequence. How should these measurements be identified in the classification? Since we have no data or statistics for these measurements it seems that we have no way to deal with them! Here we must draw upon our knowledge of the structure of reality. We know that in any environment if a measurement m is made of a small-area patch belonging within category c_i, then it is likely to make measurements $m + \delta$ for other small-area patches which also belong within category c_i. If a measurement m is typical of category c_i, then for small δ, $m + \delta$ is also typical of category c_i. Similar or close measurements are usually associated with similar or the same categories. Thus in the classification we can identify a measurement m, which did not occur in the data sequence, with the category associated with m', its nearest neighbor.

The part of the classification f* which was defined by means of the statistics generated by the experiment is called a Bayes Classification and hence the name "Bayesian approach." The part of the classification which is not Bayesian is said to be defined by a nearest neighbor search.

Acknowledgement: This work was supported by Project Themis (USAETL Contract DAAK02-68-C-0089, ARPA order No. 1079).

PART II

INPUT DATA DECK

II. INPUT DATA DECK

The data for this program are received as a sequence of measurements of small-area patches or objects with each measurement made by a sensor or set of sensors. A measurement may be, for example, the average backscatter power return from a small-area patch at incidence angles 5°, 10°, 15°, 20°, 30°, 40°, 50°, and 60°. In this case, each measurement has 8 components. The patches themselves are examined and identified as belonging within one of several given categories. The sequence of such identifications is called the "ground truth identification" or simply the "ground truth." The Bayes program can determine a Bayes classification of measurement space, based on the data and the ground truth for the data. Once a classification is determined, the Bayes program can identify each measurement within a sequence of measurements. This identification is done by a nearest-neighbor search.

The input deck is organized into four sections: title, program options, parameter cards, and format and data.

I. Title

A) The title section consists of a single card specifying the name of the data. The title may begin in column one and continue through column eighty.

II. Program Options

A) The program option section consists of two cards, the first card specifying all the input options and the second card specifying all the output options.

B) Each option is a six-character abbreviation or code.

C) The options start in column 16, are separated only by commas (no embedding blanks) and may appear in any order.

D) Only input options may appear on the input card and only output options may appear on the output card.

1) The input options are: PHOPTS, CORPTS, FLTING, PATERN, DIAGON, HLFNHF, ABSQNT.

2) The output options are: ALTICH, STDPNT, DPUNCH, PHOUT1, PHOUT2, TERMNL.

III. Parameters

A) The parameter section consists of cards, the number of which varies with the options chosen.

B) There are seven basic types of information which can possibly appear in the parameter section: gain matrix, dimensionality of measurements, number of measurements in the data set, number of categories in the classification, display size, number of levels to which the measurements will be quantized, and means of estimating the *a priori* probability distribution.

IV. Format and Data

A) There are two ways to organize the format and data: photographic form and corresponding-point form. Depending on the options chosen, the ground truth identification and its format may or may not be present. One and only one of the two forms must be specified: otherwise, an error message and termination of the job will result.

1) In the photographic form the data are organized as follows:

a) format for identification (if any)

b) identification (if any) for measurement one, measurement two,...,measurement N.

c) format for measurements

d) component one, measurement one; component one, measurement two;...;component one, measurement N; component two, measurement one; component two, measurement two;...;component two, measurement N;...component M, measurement one; component M, measurement two;...;component M, measurement N.

2) In the corresponding-point form the data are organized as follows:

a) format for identification (if any) and measurements

b) identification (if any) for measurement one; component one, measurement one; component two, measurement one;...; component M, measurement one: identification (if any) for measurement two; component one, measurement two; component two, measurement two;... component M, measurement two:... identification (if any) for measurement N; component one, measurement N; component two, measurement N;... component M, measurement N.

We now describe the options.

I. Input Options

A) PHOPTS -- is the abbreviation for photographic form, and is used when the data are in that form.

B) CORPTS -- is the abbreviation for corresponding point form, and is used when the data are in that form.

C) FLTING -- is the abbreviation for floating point, and is used when the data are punched on cards in floating-point form. It is not used when the data are punched on cards in integer form.

D) PATERN -- is the abbreviation for pattern classification by Bayes' strategy. Use of this option will output a probability matrix where element (i, j) is the conditional probability that a measurement which was identified as within the i^{th} ground truth category is identified in the classification as within the j^{th} category. A punched deck of the compacted quantized measurements with their identifications in the Bayes' classification will also be produced.

E) DIAGON -- is the abbreviation for diagonal gain matrix with ones on the diagonal. Specification of DIAGON will internally generate an identity matrix for the gain matrix. This relieves the user of the need to supply the appropriate cards in the parameter section.

F) HLFNHF -- is the abbreviation for half and half. Specification of HLFNHF will divide the data into halves: even and odd points. The first, third,. . ., data points are used to construct a Bayes' classification, and the second, fourth, . . .,data points are identified on the basis of the classification.

G) ABSQNT -- is the abbreviation for absolute maximum quantization. Specification of ABSQNT will quantize the measurements by determining the minimum and maximum values occurring among all the components, normalizing each component by subtracting this minimum, dividing by this maximum minus this minimum, and multiplying by the number of quantized levels desired. If ABSQNT is not specified, the program finds the maximum and minimum for each component, and normalizes component by component.

II. Output Options

A) ALTPCH -- is the abbreviation for alternate punch. Specification of ALTPCH produces a punched quantized data deck in the alternate form. If the data are in photographic form, the punched deck will be in corresponding-point form. If the data are in corresponding-point, the punched deck will be in photographic-point form.

B) STDPNT -- is the abbreviation for standard identification print out. Specification of STDPNT will print out the ground truth identification.

C) DPUNCH -- is the abbreviation for punched deck. Specification of DPUNCH will produce a punched quantized data deck in the same form as the input data.

D) PHOUT1 -- is the abbreviation for photograph output. Specification of PHOUT1 will identify data according to a given classification. The classification can be internally generated by the Bayes routine or it can be externally supplied in the data deck. Identification is done by a

table look-up procedure. If the quantized measurement cannot be found in the classification, a nearest-neighbor search is initiated, and the measurement is identified with the same identification as the closest measurement to it in the classification.

If the input includes ground truth identification for a data set and the data are further identified relative to a classification (by specification of PHOUT1), then a contingency table of the ground truth identification versus the classification identification will be printed out. The $(i,j)^{th}$ element of the table is the number of measurements which were identified in the i^{th} ground truth category and classified in the j^{th} ground truth category.

E) PHOUTD -- is the abbreviation for photograph output deck. Specification of PHOUTD will produce a punched deck of the identification which the photograph output routine determined. PHOUTD can only be specified if PHOUT1 is specified.

F) TERMNL is the abbreviation for remote terminal format. Specification of TERMNL will format all printed output from the Bayes program so that each line has no more than seventy columns. If TERMNL is not specified, each line is printed with one hundred thirty columns.

The generality of the program requires that the input parameters be quite variable, depending on the type of data to be processed and the options desired. For example, the PATERN option may or may not require the gain matrix as input depending on whether or not DIAGON was also specified. To facilitate setting up the input data deck, a flow chart is illustrated in Figure 1a and 1b.

The flow chart contains two symbols. The first symbol is the elongated circle (rectangle with rounded edges), which asks the question printed in the circle and requires a "true" or "false" answer. Depending upon the user's answer, one branch is chosen from the bottom of the circle.

The second symbol is the rectangle, which usually signals the addition of a single data card. Data cards are added when rectangles are encountered in the flow chart. Two exceptions to the "one-rectangle, one-data-card" rule are the gain matrix and the data itself. If the gain matrix is required (PATERN is specified and DIAGON is not specified), the user must supply all the elements for the matrix. This will take up more than one card if there are more than two ground truth identification groups. Similarly, if there are more than eight measurements in the data set, the data deck will have more than one card.

The final control card is the STOP card and is the last card in the input deck. The four-character word STOP is punched in columns one through four. The program is designed to handle more than one data set per run, and the user may place behind the first (or second, or third, etc.) data set the NAME of the next data set, the proper /INPUT and /OUTPUT cards, and the other necessary parameter cards as specified by the flow chart. The STOP card is placed after the final set of data, and informs the system to terminate the job. Excluding the STOP card or misplacing any control cards will cause a read error and termination of the job.

FLOW CHART FOR SETTING UP INPUT DATA DECK

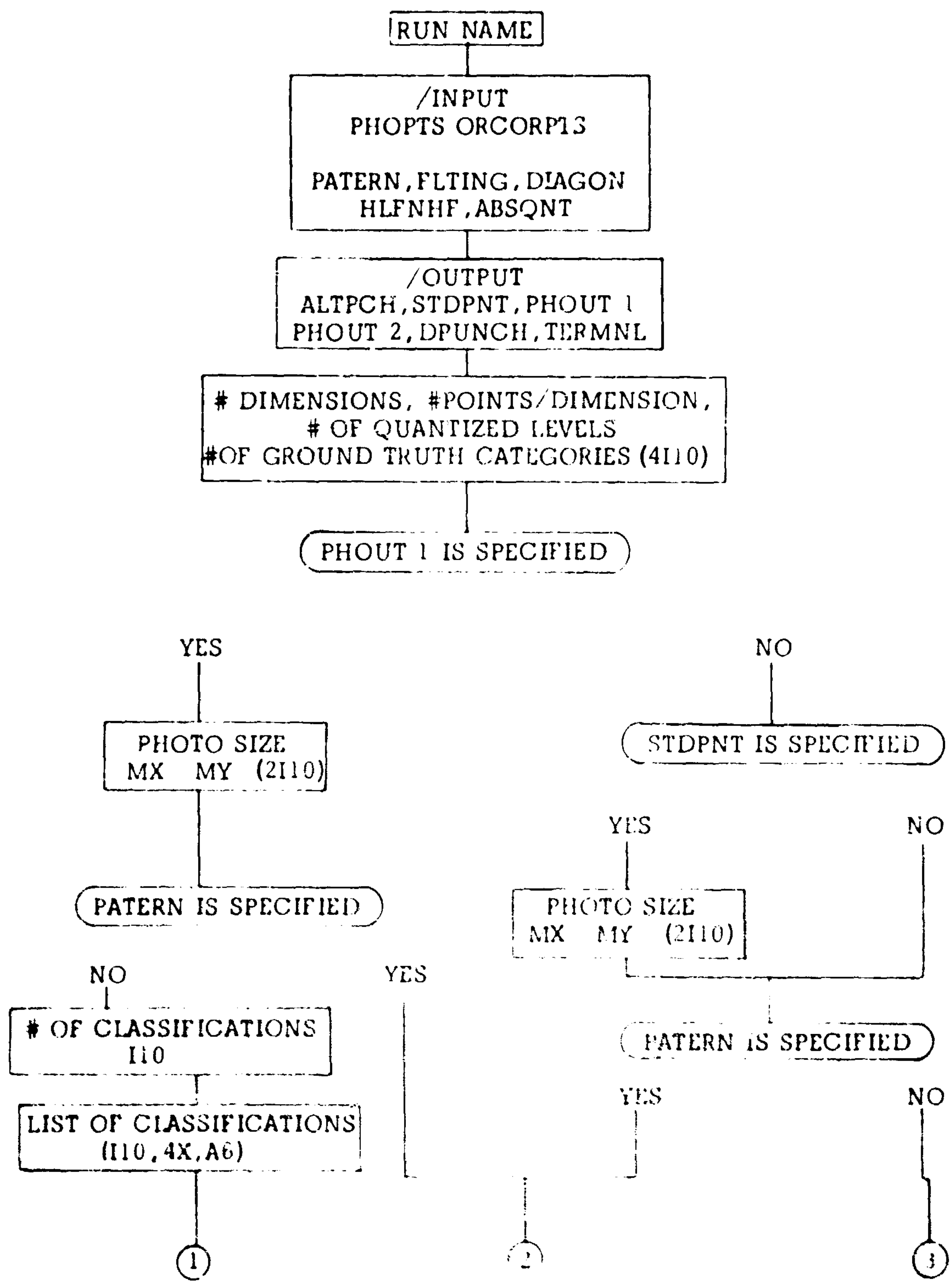


Figure 1a.

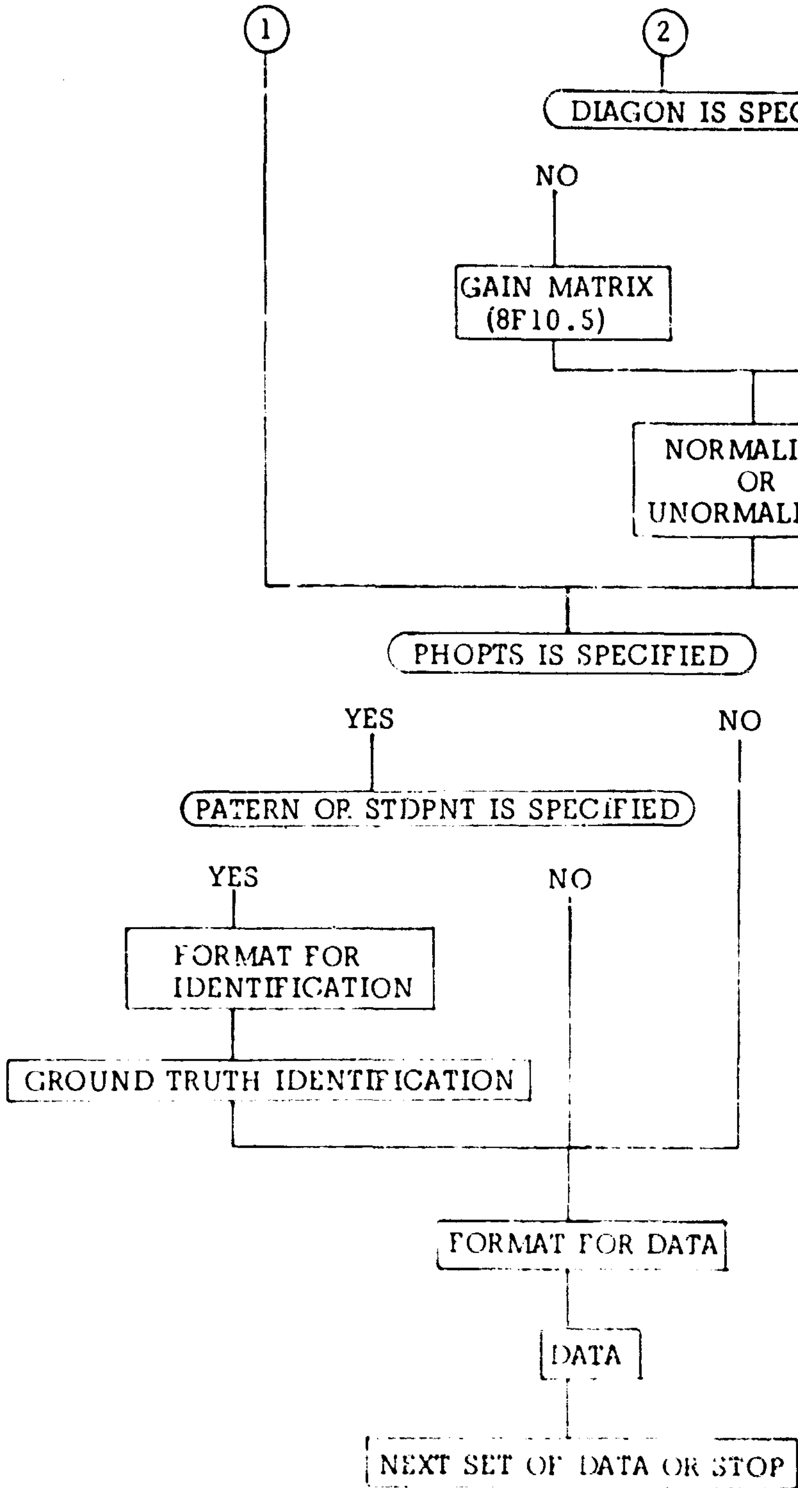

1
2
3
DIAGON IS SPECIFIED
NO
YES
GAIN MATRIX
(8F10.5)
NORMALIZED
OR
UNORMALIZED
PHOPTS IS SPECIFIED
YES
NO
PATERN OR STDPNT IS SPECIFIED
YES
NO
FORMAT FOR
IDENTIFICATION
GROUND TRUTH IDENTIFICATION
FORMAT FOR DATA
DATA
NEXT SET OF DATA OR STOP

Figure 1b.

PART III

FILE USAGE AND SUBROUTINE ORGANIZATION

III. FILE USAGE AND SUBROUTINE ORGANIZATION

Blank common storage carries all problem parameters and user options (CORPTS, PHOPTS, etc.), as well as providing a 24,000-word scratch area. Many routines in different links require such parameters as the number of dimensions of the current problem, the number of points being processed, and the number of ground truth categories.

All other communications between links are handled by tape, disc, or drum files. The program requires nine files -- 01, 02, 03, 04, 09, 10, 11, 20, 21 — as well as the input (05), output (06), and punch (43) files normally used in FORTRAN. Figure 2 describes file usage, and Figure 3 illustrates how much storage is needed on each of the files.

The Bayes program, as mentioned before, requires 36,000 words of storage in the computer core, of which 25,000 may be shared during loading. The program has been observed to process 350 sixteen-dimensional data points in less than ten minutes' processor time.

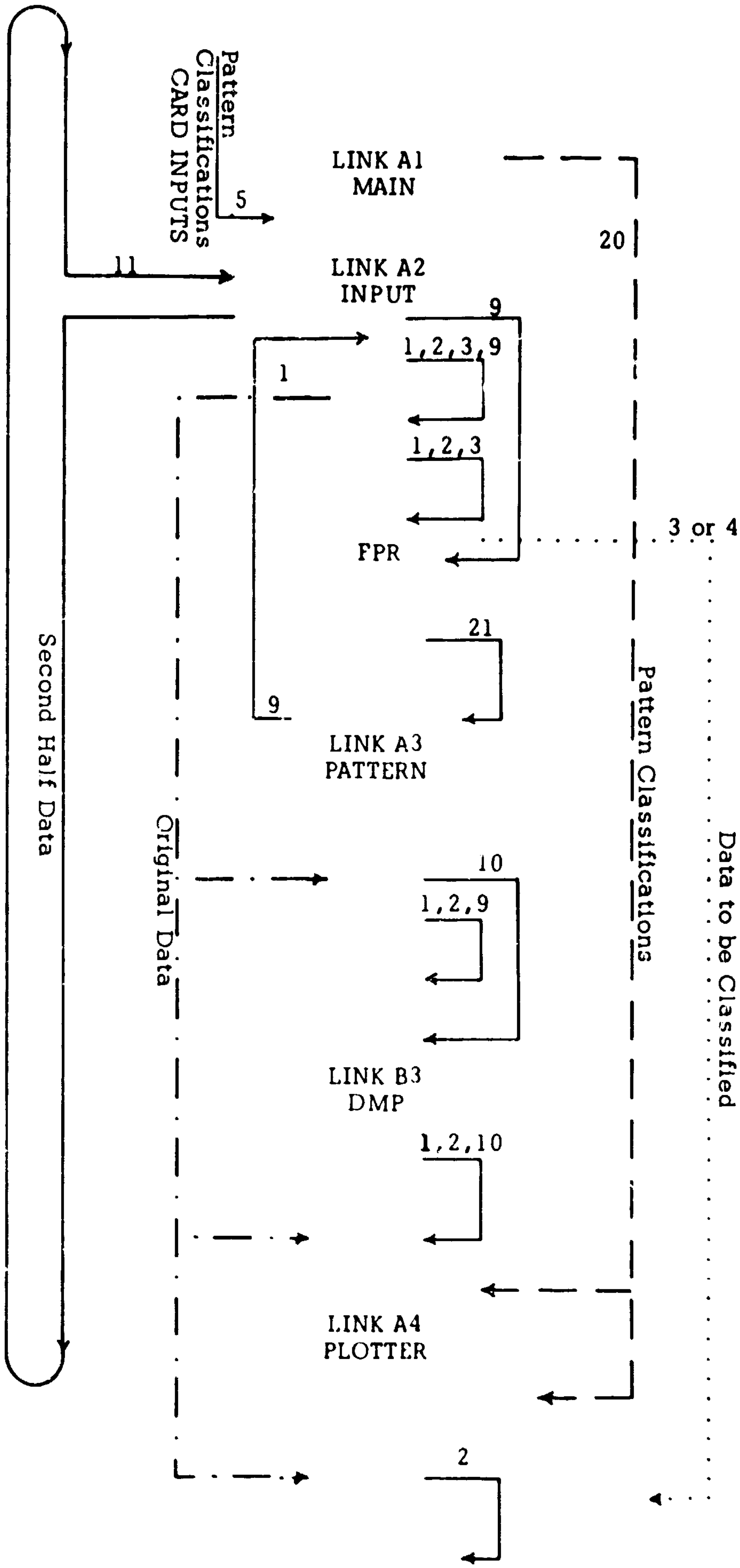


Figure 2. Tape Usage

FILE	NUMBER OF WORDS NEEDED
01	Total points
02	Total points
03	(Total Points) * # dimensions
04	
09	(Total Points) * # dimensions
10	2*(total points)
11	1/2*(total point)
20	2*(# unique n-tuples in data)
21	(Total Points) * # dimensions

Figure 3. Storage Requirements for the Tape or Disc Files

The program organization is briefly described below. Listed first are the mainline and CHNXT, the system supervisor, followed by the six-character link names used during overlay processing. Under each link name are listed the routines contained in the link. Link names containing the alphabetic character "A" refer to links essential for proper data processing: link names containing the alphabetic character "B" refer to links which output intermediate calculations, but which do not contribute to the overall program results.

LINKB2

.....(mainline) -- reserves all common storage, provides entry point to program, and contains comment cards stating program parameters and input deck setup.

CHNXT -- a small resident program used to control the entry and exits of the different links. This routine must comply with the overlay rules of the operating system in use.

LINKA1

MAIN -- parameter input and selection is accomplished in this routine. Also, if the photo classifications and the gain matrix must be read in, it is done in MAIN.

FORM -- selects and places in common all formats affected by terminal or non-terminal use.

LINKA2

INPUT -- performs data input, saves every even point for later processing if desired, and changes data to alternate form if called for.

FPR-FPR1 -- searches data for the maximum and minimum points so that proper quantization and shifting may be done in INPUT.

DEF -- defines the single-character symbols to be used in the classification.

TRANE -- prints a cross-reference between the single-character symbols used in the program and the original symbols.

CHANGE -- an assembly language routine which creates a suitable output format for the data.

LINKA3

PATERN -- Bayes program

OUTP -- prints out results of PATERN in eye-appealing format.

DECSON -- selects and assigns to each n-tuple the proper classification according to Bayes theory.

LINKB3

DMP -- used for debugging; prints out n-tuples vs. categories and n-tuples vs. classification.

LINKA4

PLOTER -- classifies and plots input data according to n-tuple classifications.

SEARCH -- searches a list of n-tuples to find the list element which is closest in distance to another n-tuple.

IDIST -- calculates n-dimensional space distances.

PART IV

EXAMPLE PROBLEM AND PROGRAM LISTING

IV. EXAMPLE PROBLEM

Suppose the problem is to input a set of data in photographic form, punch out the data in the alternate form, print the data identification, and then classify a new set of data. Let there be nine points in the horizontal direction and ten points in the vertical direction for the first data set. The data appear in Figure 4a. For the first part of the problem, the program must produce an alternate data deck, a print-out of the identification, and a Bayes' classification based on the first set of input data.

The first card is the title card. The next two cards specify the input and output options needed. The options start in column 16 and are separated by commas. The data are in photographic form, so the input card is:

```
/INPUT          PHOPTS,PATERN
```

The output card is:

```
/OUTPUT         ALTPCH,STDPNT
```

The parameter card follows. From the flow chart we see that the fourth card must specify the number of dimensions per measurement (number of photographs), total number of measurements, number of quantized levels, and number of ground truth categories. In our example the number of photographs is two and the number of measurements is ninety. We wish to quantize the data to ten levels and there are two ground truth categories. The fourth card thus appears as:

```
          2          90          10          2
```

Since PHOUT1 is not specified and STDPNT is, the fourth card must specify the photograph size, which is nine points horizontal by ten vertical. The fifth card appears as:

```
          9          10
```

111151111
111151111
111151111
111151111
555555555
555555555
111151111
111151111
111151111
111151111
111151111

Photo 1 for Test 1

222262222
222262222
222262222
222262222
333333333
333333333
222262222
222262222
222262222
222262222

Photo 2 for Test 1

Figure 4a. Data for Test 1

51115
15151
11511
15151
51115

Photo 1 for Test 2

32223
26262
22322
23232
62226

Photo 2 for Test 2

Figure 4b. Data for Test 2

Since PATERN is specified and DIAGON is not, the user must supply the gain matrix. Suppose we choose a gain matrix where we gain ten for a correct decision and lose five for an incorrect decision, as illustrated in Figure 5.

$$\begin{matrix} 10 & -5 \\ -5 & 10 \end{matrix}$$

Figure 5. Gain Matrix

The sixth card, specifying the above matrix, is

10. -5. -5. 10.

We must now indicate how the a priori probability distribution is estimated. We choose to suppose that each identification group has equal probability. Therefore NORMALIZED is specified on the next card,

NORMALIZED

The data are in photographic form, so the format for the ground truth identification must come next. In our example this would be:

(9A1).

After the identification format, the identification itself comes:

AAAABAAAA
AAAABAAAA
AAAABAAAA
AAAABAAAA
BBBBBBBBB
BBBBBBBBB
AAAABAAAA
AAAABAAAA
AAAABAAAA
AAAABAAAA

Finally we reach the format for the data, the data itself, and the STOP card. In our example these appear as:

```
(9I1)
111151111
111151111
111151111
111151111
555555555
555555555
111151111
111151111
111151111
111151111
111151111
222262222
222262222
222262222
222262222
333333333
333333333
222262222
222262222
222262222
222262222
STOP
```

The input data deck is illustrated in Figure 5.

At this point we must run a job with the input deck as shown in Figure 6. The job produces a punched deck of the Bayes classification. We must now identify a new set of data which is illustrated in Figure 4b. This is done by a separate job. The first card is, as usual, the title card. The second and third cards are the input and output option cards. The new set of data is in photographic form, and we wish to have a print-out of the identification for it, based on the classification of the previous job. The next cards thus appear:

```
/INPUT          PHOPTS
/OUTPUT         PHOUT1
```

There are two dimensions, twenty-five measurements, ten quantized levels, and two identification groups. The fourth card is:

```
2         25         10         2
```

Since PHOUT1 is specified, the next card must indicate the photographic size which is, in our example, five points horizontal by five vertical. The next card is thus:

```
5         5
```

The number of quantized measurements in the classification and the Bayes classification itself come next. These cards were obtained from the output of the previous job.

For our example, they are:

```
3
          65.02      35.02      21.01
```

All that now remains is the format for the data, the data itself, and the STOP card.

```
(5I1)
51115
15151
11511
15151
51115
32223
26262
22322
23232
62226
STOP
```

The input data deck is illustrated in Figure 7.

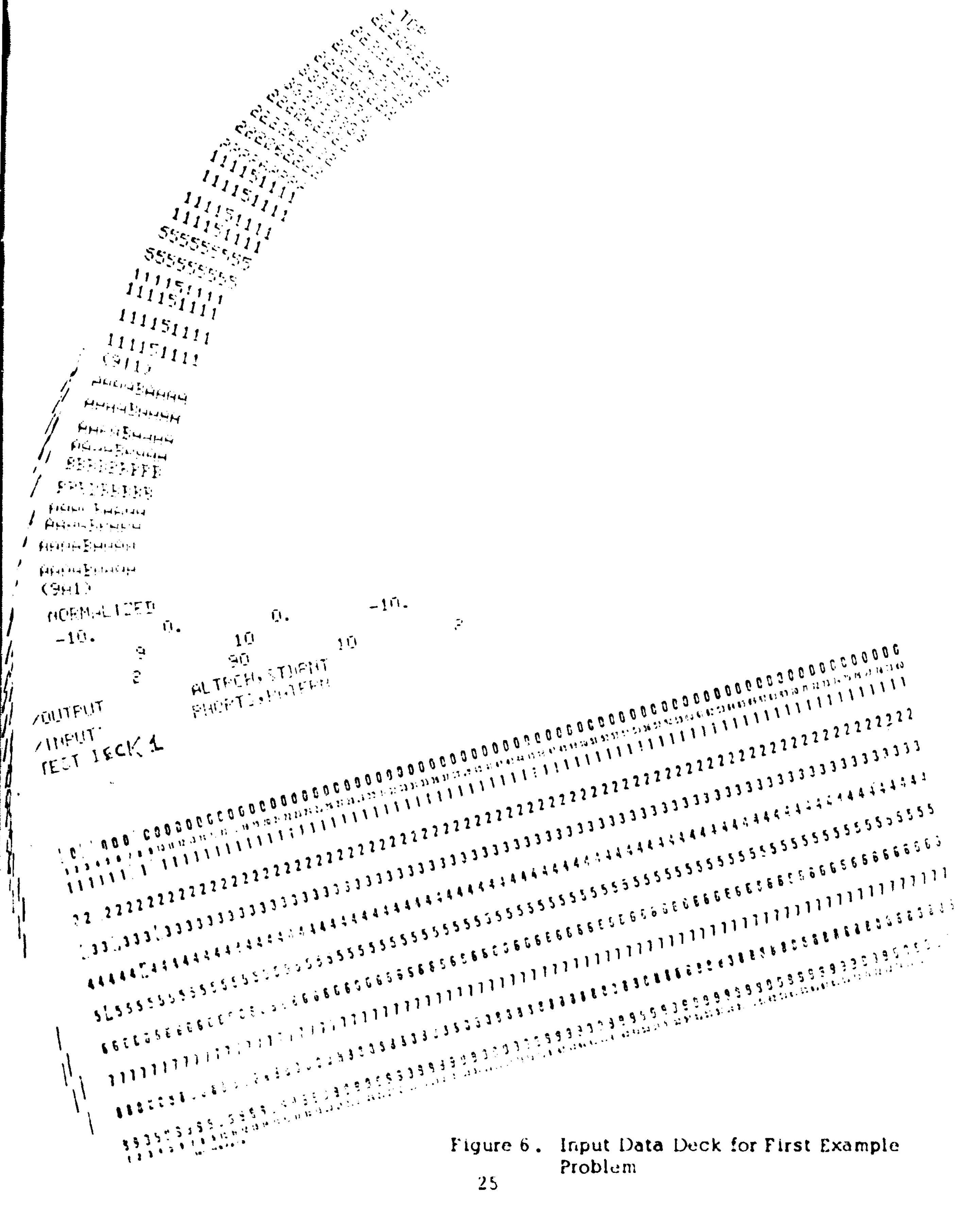


Figure 6. Input Data Deck for First Example Problem

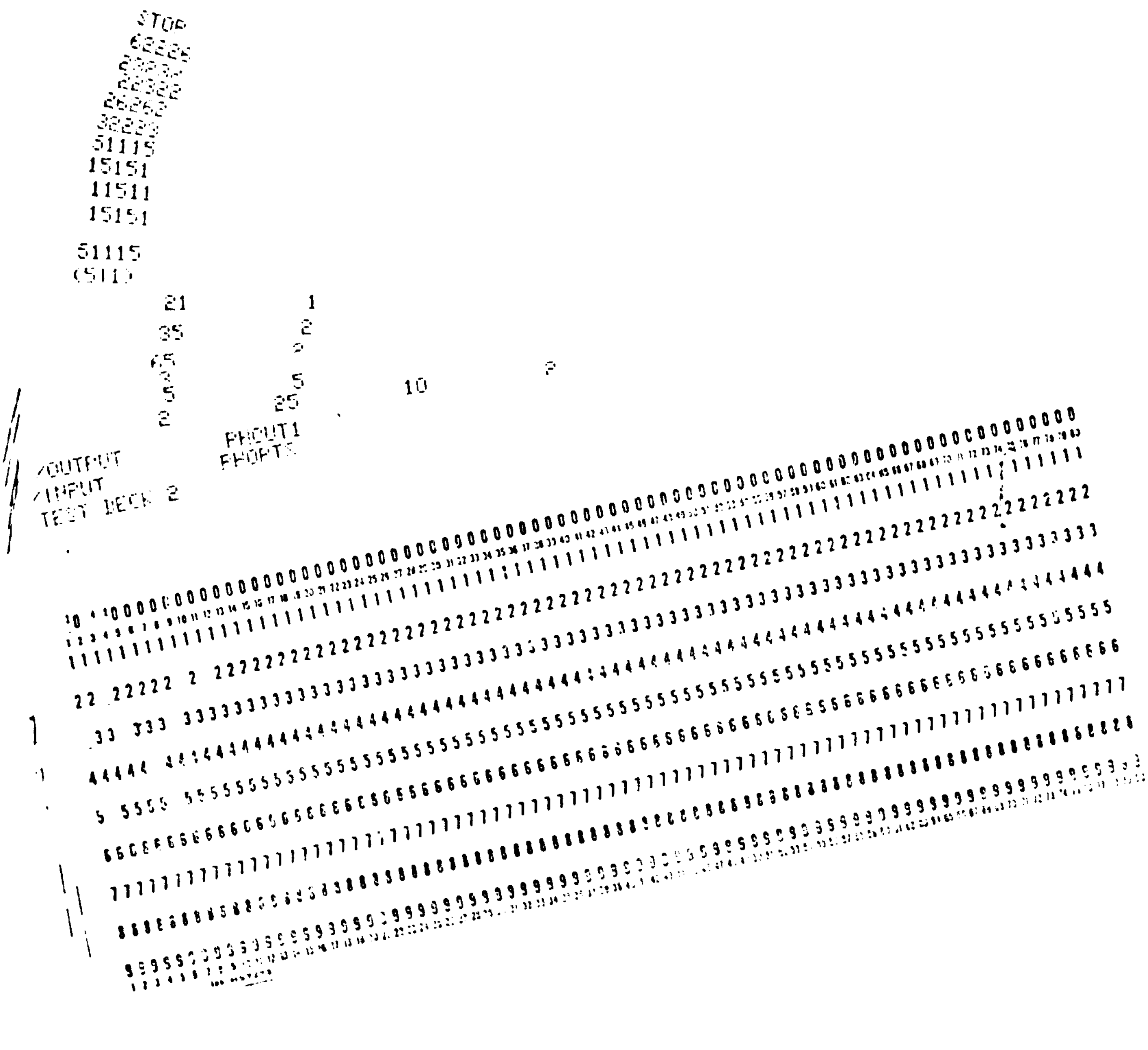


Figure 7. Input Data Deck for Second Example Problem

TEM DATA SPECIFICATIONS

```
C                 ***
C     SYSTEM DATA SPECIFICATIONS
C
C    ....................................................................
C
C    EACH DATA SET SHOULD BE PREFIXED WITH A SINGLE  CARD
C      CONTAINING THE RUN NAME
C
C     1 ST CARD  (INSTRUCTION PARAMETERS) FORMAT(5(A6,1X))
C        THIS CARD CONTAINS UP TO FIVE SPECIFICATIONS SEPERATED BY A
C        COMMA ENABLING THE USER TO ENTER AND EXIT THE PROGRAM AT
C        ANY POINT DESIRED.
C
C        THE PROGRAM EXPECTS THE INPUT DATA IN ONE OF TWO FORMS, THE
C        USER IS RESPONSIBLE FOR SPECIFING THE FORM.  IF NEITHER OR
C        BOTH ARE SPECIFIED AN ERROR MESSAGE RESULTS.
C
C        LITERAL          DATA FORM
C
C        PHOPTS           TRAINING REGIONS,PHOTO 1 ' PHOTO 2 ' ' ETC.
C
C        COMPTS           POINT1-PHOTO 1 , POINT1-PHOTO 2, ----------
C                         POINT2-PHOTO 1, POINT 2-PHOTO 2, ----------
C                           .
C                           .
C                         POINT N-PHOTO 1, POINT N-PHOTO 2, ---------
C
C        THE OTHER PARAMETERS THAT MAY BE SPECIFIED ARE
C
C        LITERAL
C
C        ALTPCH           A DATA DECK WILL BE MADE IN THE ALTERNATE D
C                         'DATA FORM'
C
C        STDPNT           A POINT OUT OF THE TRAINING REGIONS WILL BE
C                         MADE
C
C        PATERN           SUBROUTINE PATERN WILL BE CALLED
C                         (SCATTEROMETRY PROGRAM) A TAPE SUCH AS IS
C                         MADE IN PATTAP MUST BE AVAILABLE ON FORTRAN
C                         UNIT 1
C
C        PHOUT1           PREPARES A PHOTO PRINT USING PATERN
C                         CLASSIFICATION ( IF PATERN IS NOT SPECIFIED
C                         PATERN CLASSIFICATIONS MUST BE READ IN-
C                         SEE 3RD CARD )
C
C        PHOUT2           PREPARES A PHOTO PUNCHED OUTPUT USING
C                         PATERN CLASSIFICATIONS
C
C        SPUNCH           PUNCH OUTPUT OF QUANTIZED DATA
C
C        FLTING           SPECIFIED IF DATA IS IN FLOATING POINT
C                         FORMAT
C
C       TERMNL             SPECIFIED IF REMOTE TERMINAL IS BEING USED
C
C        DIAGON           SPECIFIED IF THE GAIN MATRIX USED IN PATERN
C                         IS TO HAVE ONES ON ALL DIAGONAL ELEMENTS
C                         AND ZEROS ELSEWHERE
C
C    ....................................................................
C
C     2 ND  CARD
C
C        NO. PHOTOS,NO. POINTS PER PHOTO,
C                         DESIRED LEVEL OF QUANTIZATION, NO. OF
C        TRAINING REGIONS (NR), FORMAT(5I10)
C        NR MUST BE LESS THAN 26
C    ....................................................................
C
C
C     3 RD   CARD / CARDS
C
C        IF PHOUT1 IS SPECIFIED, READ IN PHOTO SIZE
C              (NA,NB) FORMAT(2I10) )
C        AND PATERN CLASSIFICATIONS IF SUBROUTINE PATERN IS NOT USED.
C        NCLAS = NO. OF POINTS CLASSIFIED FORMAT(I10), NTUPLES WITH
C        CLASSIFICATIONS FORMAT(8F10.0)
C
C     IF STDPNT IS SPECIFIED, READ IN PHOTO SIZE (NX,NY) FORMAT(2I10)
C        IF PATERN IS SPECIFIED,
C          READ IN THE GAIN MATRIX TO BE USED (NRXNR) FORMAT(8F10.5)
C          READ IN 'NORMALIZED' OR 'UNNORMALIZED' FORMAT(A6,A6) LEFT-
C          JUSTIFIED
C
C    ....................................     ..........................
C
C     OTHERS
C
C        READ FORMAT FOR DATA
C     READ DATA
C
C        PUNCH STOP IN COL 1-4 OF THE LAST CARD - THIS ENABLES MORE
C        THAN ONE SET OF DATA TO BE PROCESSED ON A SINGLE RUN
C
C    ....................................................................
C
C                 ***
      COMMON NPOL,FORMT(10)
      COMMON F
      [illegible]
     [illegible]
      COMMON L0,L9,L10,L11,L12,L13,L14
      [illegible]
     [illegible]
      LOGICAL [illegible]
      LOGICAL [illegible]
      INTEGER F
C      SET ACTN WORD IN COMMON TO 0 AND TAX WORD IN COMMON TO 6
      F=3
      N=0
      CALL EXEC
      STOP
      END
```

[illegible] WORDS OF MEMORY USED BY THIS COMPILATION

17564 01 05-02-69   10.141

```
      SUBROUTINE CHNXT
      DIMENSION NAME(3)
      COMMON MPO1,FORMT1(12)
      COMMON F
      COMMON FMT1(15),FMT2(3),FMT9(5),FMT11(6),FMT12(7),FMT13(3),FMT14(2
     1),FMT16(2),FMT22(3),FMT56(6),FMT57(4),FMT58(8)
      COMMON L8,L9,L10,L11,L12,L13,L14
      COMMON N,M,LEVEL,KZ1,KCOUNT(36),ITAPE,AMAT(625),LT,L1,L2,L3,L4,L5,
     1L6,KK,MX,MY,K,L7,MA,MB,MC,ZZ(24000)
      LOGICAL LT,L1,L2,L3,L4,L5,L6,L7
      LOGICAL L8,L9,L10,L11,L12,L13
      INTEGER F
      DATA NN/6HSTOP  /
C                             ...
C      K POINTS TO LINK IN USE
C                             ...
    1 K=K+1
      GO TO (2,3,4,5,6,7), K
    2 READ (F,9) NAME
      IF (NAME(1).EQ.NN) STOP
C      CALL MAIN
      CALL LINK (6HLINKA1)
C      CALL IMPUT
    3 CALL LINK (6HLINKA2)
    4 IF (L6) GO TO 6
      CALL LINK (6HLINKA3)
    5 IF (L6) GO TO 6
      CALL LINK (6HLINKB3)
    6 K=5
      IF (L7) GO TO 7
      CALL LINK (6HLINKA4)
    7 IF (L12) GO TO 8
      K=0
      GO TO 1
C                             ...
C        IF DATA IS RUN IN HALFS SET UP PARAMETERS
C                             ...
    8 L12=.FALSE.
      L13=.TRUE.
      L5=.FALSE.
      L6=.TRUE.
      L7=.FALSE.
      M=MPO1/2
      K=1
      WRITE(6,100) M
  100 FORMAT(//34H SECOND HALF, NUMBER OF POINTS IS ,I10)
      GO TO 1
      ENTRY MNAME
C                             ...
C     PRINT ID NAME
C                             ...
      WRITE (6,FMT22) NAME
      RETURN
C                             ...
C
C                             ...
    9 FORMAT (3A6)
      END
```

<><><><><><><><><><><><><><><><><><><><><><><><><><><><><><

\-   22986 WORDS OF MEMORY USED BY THIS COMPILATION

<><><><><><><><><><><><><><><><><><><><><><><><><><><><><

R0370 01 09-29-79 21.004 CMDEV EXECUTION REPORT

```
$       FORTRAN HTKCK,COMDH
      SUBROUTINE MAIN
C      LINMAI
      DIMENSION LL(14), PAR(12), CH(14)
      COMMON MPO',FORMT1(12)
      COMMON F
      COMMON FMT1(19),FMT2(3),FMT8(9),FMT11(6),FMT12(7),FMT13(3),FMT14(2
     $),FMT16(2),FMT22(3),FMT56(8),FMT97(4),FMT99(8)
      COMMON LA,LV,L10,L11,L12,L13,L14
      COMMON NP,M,LEVEL,K71,KCOUNT(34),ITAPE,AMAT(425),LT,L1,L2,L3,L4,L9
     $,L6,NN,MY,MV,NDUMY,L7,NA,NB,NC,ZZ(24028)
      LOGICAL LL,L1,L3,L4,L9,L2,L6,LT,L7
      LOGICAL L8,L9,L10,L11,L12,L13,L14
      LOGICAL SS
      INTEGER F
      REAL A(8),H(P)
      DATA CH/ANCOMPTS,AMPHOPTS,AMALTPCH,AHSTOPNT,AHPATTAP,AHPATERN,AHPH
     1OUT1,AHFLTING,AHPHOUT2,AHPUNC4,AHTERMNL,AHDIAGON,AHNLFHMF,AHARBOW
     $T/
      DATA BLK/AH      /
      DATA STP/AHSTOP  /
      DATA NOR/AHNORMAL/
C                          ***
C      READ IN PARAMETERS ON /INPUT AND /OUTPUT CARDS
C                          ***
      READ (F,21) PAR
      WRITE(6,21) PAR
      IF (PAR(1).EQ.STP) GO TO 3
C                          ***
C      INITIALIZE LL
C                          ***
      DO 1 I=1,14
    1 LL(I)=.TRUE.
C                          ***
C      SEEK PARAMETERS IN LIST
C                          ***
      DO 2 I=1,12
      IP=I
      SS=.TRUE.
      IF(PAR(I).EQ.BLK) GO TO 2
      DO 200 J=1,14
      IF(PAR(I).NE.CH(J)) GO TO 200
      LL(J)=.FALSE.
      SS=.FALSE.
  200 CONTINUE
      IF(SS) GO TO 300
    2 CONTINUE
      IF (.NOT.(LL(1).AND.LL(2))) GO TO 4
C                          ***
C      IF NEITHER PHOPTS OR COMPTS IS SPECIFIED PRINT ERROR AND ABORT
C                          ***
      WRITE (6,22)
    3 STOP
C                          ***
C      INITIALIZE L
C                          ***
    4 L1=.FALSE.
      L2=.TRUE.
      L3=.TRUE.
      L4=.TRUE.
      L5=.TRUE.
      L6=.TRUE.
      L8=.FALSE.
      L9=.TRUE.
      L10=.TRUE.
      L11=LL(11)
      L12=.FALSE.
      L13=.FALSE.
C                          ***
C      TRANSFER LL TO L
C                          ***
      L14=.FALSE.
      IF (.NOT.LL(14)) L14=.TRUE.
      IF (.NOT.LL(13)) L12=.TRUE.
C   LOAD FORMATS FOR EITHER TERMINAL OR COMPUTER USE DEPENDING ON LL(11)
      CALL FFAM(LL(11))
      IF (LL(10)) L10=.FALSE.
      IF (LL(6)) L6=.TRUE.
      IF (LL(9)) L9=.FALSE.
C      IF PATERN IS SPECIFIED CONTINUE
      IF (LL(9)) GO TO 5
      L3=.FALSE.
      L2=.FALSE.
      L6=.FALSE.
    5 IF (LL(4)) GO TO 6
      L9=.FALSE.
    6 IF (LL(3)) GO TO 7
      L4=.FALSE.
      L3=.FALSE.
C      LL(1) FALSE MEANS COMPTS WAS SPECIFIED
C      L1 TRUE MEANS COMPTS WAS SPECIFIED
    7 IF (LL(1)) GO TO 8
      L1=.TRUE.
C                          ***
C      READ NO. OF PHOTOS, NO. POINTS PER PHOTO,
C                                  DESIRED LEVEL OF QUANTIZATION,
C        AND NUMBER OF TRAINING REGIONS
C                          ***
    8 READ (F,23) NP,M,LEVEL,NN
      WRITE(6,23) NP,M,LEVEL,NN
      MPE1=M
C      L7 FALSE MEANS PHOUT1 WAS SPECIFIED
      L7=.FALSE.
      IF (LL(7)) L7=.TRUE.
```

```
      IF (L(7)) GO TO 12
C                          ***
C     READ IF PHOTO SIZE MH-HORIZONTAL  MV-VERTICAL
C                          ***
      READ (5,24) MH,MV
      WRITE(6,24) MH,MV
      MH=MH
      MV=MV
      MPOINT=1
      IF(MPOINT.LE.M) GO TO 30
   31 CONTINUE
C      IF PATERN IS SPECIFIED GO TO 12
      IF (.NOT.LL(8)) GO TO 12
      MC=0
C                          ***
C     IF PATERN NOT USED READ IN N-TUPLES AND CLASSIFICATIONS
C     STORING BY EIGHTS ON FILE 20
C                          ***
      READ (5,25) NCLAS
      REWIND 20
      J=NCLAS/8
      IF (J.EQ.0) GO TO 10
      DO 9 I=1,J
      K=8
      WRITE (20) K
      MC=MC+1
      READ (5,26) (A(M),B(M),M=1,8)
      WRITE (20) (A(M),M=1,8)
    9 WRITE (20) (B(M),M=1,8)
   10 J=NCLAS-8*J
      IF (J.EQ.0) GO TO 11
      WRITE (20) J
      MC=MC+1
      READ (5,26) (A(M),B(M),M=1,J)
      WRITE (20) (A(M),M=1,J)
      WRITE (20) (B(M),M=1,J)
   11 J=0
      WRITE (20) J
C                          ***
C        .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
C                          ***
      REWIND 20
C                          ***
C      IF STRPNT IS SPECIFIED , READ IN THE PHOTO SIZE (MH,MV)
C                          ***
   12 IF (L(5)) GO TO 14
      IF (L(7)) GO TO 13
      MH=MH
      MV=MV
      GO TO 14
   13 READ (5,27) MH,MV
      WRITE(6,27) MH,MV
      MPOINT=2
      IF(MPOINT.LE.M) GO TO 30
   32 CONTINUE
C                          ***
C      IF PATERN IS SPECIFIED, READ IN THE LOSS MATRIX TO BE USED
C        (KK X KK )
C                          ***
   14 IF (L(6)) GO TO 20
      KK=KK**2
C      KK IS THE NUMBER OF TRAINING REGIONS SQUARED
C      IF DIAGON IS SPECIFIED SET UP DIAGONAL GAIN MATRIX
      IF (LL(10)) GO TO 17
      DO 15 I=1,625
   15 AMAT(I)=0.0
      DO 16 I=1,KK
      II=(I-1)*KK+I
   16 AMAT(II)=1.00
      GO TO 18
   17 READ (5,28) (AMAT(I),I=1,KK)
   18 DO 19 I=1,KK
   19 AMAT(I)=-AMAT(I)
      LT=.FALSE.
C                          ***
C     READ IN NORMALIZED OR UNNORMALIZED
C     LT TRUE SIGNALS NORMALIZED
C                          ***
      READ (5,29) NORM
      IF (NORM.EQ.NOR) LT=.TRUE.
   20 CALL CHART
      STOP
C                          ***
C
C                          ***
   21      -T (1X8,A110,A8)/1X8,A110,A8))
   22 FORMAT (1X,20HINCORRECT PARAMETER CARD)
   23 FORMAT (4I10)
   24 FORMAT (2I10)
   25 FORMAT (I10)
   26 FORMAT (7(I10,I10))
   27 FORMAT (2I10)
   28 FORMAT (8F10.5)
   29 FORMAT (1X,A8)
   30  CONTINUE
      MPOINT=MPOINT
      WRITE(6,100) M,MH,MV,MPOIN,MPOIN
  100 FORMAT(1X[illegible]TOTAL NUMBER OF POINTS WAS PURPORTED TO BE ,I10/
     1 12H BUT MH*MV= ,I5,1HX,I5,3H = ,I10/31H SET TOTAL NUMBER OF POINTS
     2 AT ,I10)
      M=MH*MV
      GO TO (31,32),MPOINT
  300 WRITE(6,301) PAR(IP)
  301 FORMAT(1X,A8,[illegible] IS A BAD PARAMETER ON /INPUT OR /OUTPUT CARD)
      STOP
      END
```

17964 01 05-02-69   10.163

```
      SUBROUTINE FFRM(L)                                                    D   1
      COMMON MPO1,FORMT1(12)                                                D   2
      COMMON F                                                              D   3
      COMMON FMT1(15),FMT2(3),FMT9(5),FMT11(6),FMT12(7),FMT13(3),FMT14(2    D   4
     1),FMT16(2),FMT22(3),FM156(6),FMT57(4),FMT58(8)                        D   5
      DIMENSION O(64), T(64)                                                D   6
      LOGICAL L                                                             D   7
      INTEGER F                                                             D   8
      DATA T/6H(//30X,6H,11HLA,6HBLE TA,6HBLE //,6H 16X,2,6HOHPLOT,6H SY    D   9
     1MBO,6HLS VS ,6HINPUT ,6HSYMBOL,6HS  //2,6H6X,A1,,6H12X,A6,6H)         D  10
     2,6H       ,6H(/26X,,6HA1,12X,6H,A6)  ,6H(27X,1,6H6HTRAI,6HNING R,6H   D  11
     3EGIONS,6H //)  ,6H(24X,2,6HOHPHOT,6HO CLAS,6HSIFICA,6HTION /,6H/)     D  12
     4   ,6H(//15X,6H,29HTR,6HAINING,6H REGIO,6HN CLAS,6HSIFIED,6H AS /)    D  13
     5,6H(5X,8(,6H5X,I2),6H/)    ,6H(1H*,2,6HX,A1) ,6H(3X,A1,6H)      ,6H   D  14
     6(1H1,2,6H5X,3A6,6H//)   ,6H(24X,2,6H2HTHE ,6HPROBAB,6HILITY ,6HMAT    D  15
     7RIX,6H)      ,6H(6X,I2,6H,8(2X,,6HF4,2)),6H       ,6H(/////,6H//21X,  D  16
     8,6H17HTOT,6HAL PRO,6HBABILI,6HTY //2,6H3X,1E1,6H2,4)  /               D  17
      DATA O/6H(//60X,6H,11HLA,6HBLE TA,6HBLE //,5H51X,31,6HHPLOT ,6HSYM    D  18
     1BOL,6HS  VS ,6HINPUT ,6HSYMBOL,6HS //56,6HX,A1,1,6H2X,A6),6H          D  19
     2,6H      ,6H(/56X,,6HA1,12X,6H,A6)  ,6H(57X,1,6H6HTRAI,6HNING R,6H    D  20
     3EGIONS,6H //)  ,6H(54X,2,6HOHPHOT,6HO CLAS,6HSIFICA,6HTION /,6H/)     D  21
     4   ,6H(//50X,6H,29HTR,6HAINING,6H REGIO,6HN CLAS,6HSIFIED,6H AS /)    D  22
     5,6H(4X,8(,6H13X,I2,6H)/)    ,6H(1H*,1,6HX,A1) ,6H (2X,A,6H1)     ,6H  D  23
     6(1H1,5,6H5X,3A6,6H//)   ,6H(54X,2,6H2HTHE ,6HPROBAB,6HILITY ,6HMAT    D  24
     7RIX,6H )    ,6H(5X,I2,6H,8(3X,,6HE12.4),6H)      ,6H(/////,6H//51X,   D  25
     8,6H17HTOT,6HAL PRO,6HBABILI,6HTY  ./,6H/53X,1,6HE12.4)/               D  26
C                                 ...                                       D  27
C     SETUP FORMATS FOR USE IN THE PROGRAM                                  D  28
C                                 ...                                       D  29
      IF (L) GO TO 2                                                        D  30
      DO 1 I=1,64                                                           D  31     4
    1 FMT1(I)=T(I)                                                          D  32     5
      RETURN                                                                D  33     7
    2 DO 3 I=1,64                                                           D  34     8
    3 FMT1(I)=O(I)                                                          D  35     9
      RETURN                                                                D  36    11
      END                                                                   D  37-   12
```

<>
*   22987 WORDS OF MEMORY USED BY THIS COMPILATION
<>

```
00370 01  05-29-69   21.06A                    CMOEX EXECUTION REPORT
$      FORTRAN AUECK,COMDK
CIMT         INPUT LINEA2
      SUBROUTINE INPUT
C      LINEA2
      DIMENSION LARGE(20), LSL(20), ACC(20)
      DIMENSION TRAIN(8000), STO(36), KCOUNT(36), DATA(8000), NCOUNT(37)
     1 , STA(8000),FORM(12)
      COMMON NPO1,FORM1(12)
      COMMON F
      COMMON FMT1(15),FMT2(3),FMT9(5),FMT11(6),FMT12(7),FMT13(3),FMT16(2
     1),FMT18(2),FMT22(3),FMT54(4),FMT57(4),FMT58(6)
      COMMON L8,L9,L10,L11,L12,L13,L14
      COMMON N,M,LEVEL,NZ1,NCOUNT,ITAPE,KMAT(675),L7,L1,L2,L3,L4,L5,L6,K
     1R,MX,MY,NDUMP,L7,MM1,MM2,MM3,ZZ(24000)
      EQUIVALENCE (ZZ(1),STA(1)),(ZZ(8001),DATA(1)),(ZZ(16001),TRAIN(1))
      LOGICAL LOG,L7,L1,L2,L3,L4,L5,L6,L7
      LOGICAL L8,L9,L10,L11,L12,L13
      LOGICAL L14
      DIMENSION IPRIME(20)
      INTEGER F
      INTEGER DATA,STA
      DATA BLANK/1H /
      DATA IPRIME/1,2,4,5,8,10,12,16,20,24,28,30,32,36,40,44,48,60,62,90/
C                          ***
C
C       TAPE 11 STORES THE SECOND HALF OF THE TRAINING REGION AND THE
C       SECOND HALF OF THE DATA FOR H.FORM
C       TAPE 9 STORES THE ORIGINAL DATA SET BEING PRESENTLY WORKED ON
C       THIS IS THE ENTIRE DATA SET IF AN ENTIRE CLASSIFICATION IS DONE
C       OR IT IS THE FIRST HALF OF THE DATA IF HLFHMF
C
C       LEVEL= LEVEL OF QUANTIZATION
C       NZ1=NO. OF SIGNIFICENT FIG. IN INPUT DATA NZ1 .LE. 9
C       M=NO. OF DATA POINTS PER PHOTO
C       N=NO. OF PHOTOS
C
C                          ***
      IF (.NOT.L4) WRITE (45,84)
C                          ***
C     INITILIZE KCOUNT ,MPO,K,KK,ITAPE
C                          ***
      DO 1 I=1,36
    1 KCOUNT(I)=0
      IF (L13.OR.L12) REWIND 11
      MPO=M
      K=1
      KK=1
      ITAPE=1
      MM1=MX
      MM2=MY
C                          ***
C     JUMP TO INPUT SECTIONS ON TYPE OF DATA L1=TRUE SIGNALS COMPTS
C                          ***
      IF (L1) GO TO 30
C                          ***
C     IF NEITHER PATERN OR ALTERNATE PUNCH  GO TO 41
C                          ***
      IF (L5.AND.L2) GO TO 9
C                          ***
C     L13 =TRUE SIGNALS SECOND HALF DATA
C                          ***
C      L12 IS TRUE WHEN HLFHMF IS SPECIFIED AND THE FIRST HALF OF DATA IS BEING
C     RUN.    FOR THE SECOND HALF, L12 IS FALSE
      IF (.NOT.L13) GO TO 2
      READ (11) (TRAIN(I),I=1,M)
      GO TO 600
C                          ***
C     READ IN FORMAT AND DATA
C                          ***
    2 READ (F,85) FORM
      WRITE(6,101) FORM
  101 FORMAT(20H FORMAT FOR TRAINING DATA IS ,12A6)
      READ (F,FORM) (TRAIN(I),I=1,M)
      IF (L12) WRITE (11) (TRAIN(I),I=2,M,2)
      IF (.NOT.L13) GO TO 3
  600 CONTINUE
C      SECOND HALF
      MM1=MX-MX/2
      MM2=MY
      IF(MM1*MM2.EQ.MPO) GO TO 3
      DO 200 JACK=1,20
      IF(MPO/IPRIME(JACK)*IPRIME(JACK).NE.MPO) GO TO 200
      INDEX=IPRIME(JACK)
      INDEXY=MPO/IPRIME(JACK)
  200 CONTINUE
      MM1=INDEX
      MM2=INDEXY
    3 IF (.NOT.L12) GO TO 9
C      FIRST HALF
      J=1
      DO 4 I=3,M,2
      J=J+1
    4 TRAIN(J)=TRAIN(I)
      MPO=M/2
      IF (2*MPO.NE.M) MPO=MPO+1
      MM1=MX/2
      MM2=MY
      IF(MM1*MM2.EQ.MPO) GO TO 9
      DO 300 JACK=1,20
      IF(MPO/IPRIME(JACK)*IPRIME(JACK).NE.MPO) GO TO 300
      INDEX=IPRIME(JACK)
      INDEXY=MPO/IPRIME(JACK)
  300 CONTINUE
      MM1=INDEX
```

```
      NN2=INCREM
    5 NCOUNT(1)=1                                                      E  81
      STO(1)=TRAIN(1)                                                  E  82
C      STO IS THE ARRAY OF UNIQUE SYMBOLS USED IN TRAINING REGION
C      NCOUNT(I) IS THE NUMBER OF TIMES THE ITH UNIQUE SYMBOL OCCURRED
C      IN TRAINING REGION
      DO 8 I=2,M                                                       E  83
    6 IF (TRAIN(I).EQ.STO(K)) GO TO 7                                  E  84
      K=K+1                                                            E  85
      IF (K.LE.NR) GO TO 6                                             E  86
      STO(K)=TRAIN(I)                                                  E  87
      NCOUNT(K)=1                                                      E  88
      NR=NR+1                                                          E  89
      GO TO 8                                                          E  90
    7 NCOUNT(K)=NCOUNT(K)+1                                            E  91
    8 K=1                                                              E  92
C                               ...                                    E  93
C     NCOUNT CONTAINS THE NUMBER OF POINTS IN EACH TRAINING REGION     E  94
C       NR IS THE NO. OF TRAINING REGIONS                              E  95
C     L3=FALSE SIGNALS PATERN OR LINEAR                                E  96
C                               ...                                    E  97
    9 IF (L3) GO TO 76                                                 E  98
      NCOUNT(1)=0                                                      E  99
      NRP1=NR+1                                                        E 100
C                               ...                                    E 101
C     NCOUNT CONTAINS CUMULATIVE TRAINING REGIONS                      E 102
C                               ...                                    E 103
      DO 10 I=2,NRP1                                                   E 104
   10 NCOUNT(I)=NCOUNT(I-1)+NCOUNT(I-1)                                E 105
      REWIND 3                                                         E 106
      IF(L13) GO TO 11
C                               ...                                    E 108
C     READ FORMAT FOR DATA                                             E 109
C                               ...                                    E 110
      READ (5,85) FORM                                                 E 111
C                               ...                                    E 112
C     SET UP OUTPUT FORMAT                                             E 113
C                               ...                                    E 114
      CALL CHANGE (FORM,FORM1)                                         E 115
C      WRITE OUT NAME OF RUN
   11 CALL HNAME                                                       E 116
C      WRITE OUT ORIGINAL DATA
      WRITE (6,84)                                                     E 117
C                               ...                                    E 118
C       .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  E 119
      IF(.NOT.L13) WRITE(6,100) FORM,FORM1
  100 FORMAT(/27H INPUT FORMAT FOR DATA WAS ,12A6/
     1 27H OUTPUT FORMAT FOR DATA IS ,12A6)
C     READ IN DATA AND DETERMINE LARGEST AND SMALLEST DATA VALUES      E 120
C                               ...                                    E 121
      DO 12 I=1,N                                                      E 122
      LARGE(I)=0                                                       E 123
   12 LSL(I)=10000000000                                               E 124
      REWIND 9                                                         E 125
      DO 18 J=1,N                                                      E 126
      WRITE(6,777) J
  777 FORMAT(1H1,10X,6HPHOTO ,I5)
C      L13 IS TRUE IF HLFNME IS SPECIFIED AND PROGRAM IS WORKING ON
C      SECOND HALF OF DATA
      IF (.NOT.L13) GO TO 13                                           E 127
      READ (11) (DATA(I),I=1,M)                                        E 128
      GO TO 15                                                         E 129
   13 READ (5,FORM) (DATA(I),I=1,M)                                    E 130
      IF (.NOT.L12) GO TO 15                                           E 131
      WRITE (11) (DATA(I),I=2,M,2)                                     E 132
      JWRONG=1
      DO 14 I=3,M,2                                                    E 134
      JWRONG=JWRONG+1
      DATA(JWRONG)=DATA(I)
   14 CONTINUE                                                         E 137
   15 DO 606 I=1,MY
      M1S=(I-1)*MM1+1
      M1F=I*MM1
  606 WRITE(6,FORM1) (DATA(K),K=M1S,M1F)
C      L8 FALSE MEANS THAT FLTING WAS SPECIFIED
      IF (.NOT.L8) GO TO 17                                            E 140
      DO 16 I=1,MPQ
      IF (DATA(I).GT.LARGE(J)) LARGE(J)=DATA(I)                        E 142
   16 IF (DATA(I).LT.LSL(J)) LSL(J)=DATA(I)                            E 143
   17 WRITE (9) (DATA(I),I=1,M)                                        E 144
   18 CONTINUE                                                         E 145
      M=MPQ                                                            E 146
C      L8 IS TRUE IF FLTING WAS NOT SPECIFIED
      IF(L8) GO TO 190
C                               ...                                    E 147
C     IF DATA IS FLOATING USE FPQ TO DETERMINE LARGEST AND SMALLEST VALU E 148
C                               ...                                    E 149
      CALL FPQ (LSL,LARGE,N,M)                                         E 150
      GO TO 19
C                               ...                                    E 151
C       .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  E 152
C     QUANTITIZE DATA AND STORE ON TAPE 3                              E 153
C                               ...                                    E 154
  190 IF(.NOT.L14) GO TO 19
C      DATA IS TO BE ABSOLUTELY QUANTIZED
      LLARGE=LARGE(1)
      LLSL=LSL(1)
      DO 607 J=2,N
      IF(LLARGE.LT.LARGE(J)) LLARGE=LARGE(J)
      IF(LLSL.GT.LSL(J)) LLSL=LSL(J)
  607 CONTINUE
      DO 608 J=1,N
      LARGE(J)=LLARGE
  608 LSL(J)=LLSL
```

```
   19 REWIND 9
      DO 20 J=1,N
   20 ACC(J)= FLOAT(LEVEL-1)/FLOAT(LARGE(J)-LSL(J))
      DO 22 J=1,N
      READ (9) (DATA(I),I=1,M)
      DO 21 I=1,M
      DATA(I)=FLOAT(DATA(I)-LSL(J))*ACC(J)+.5
   21 CONTINUE
C      TAPE 3 HAS QUANTIZED DATA
      WRITE (3) (DATA(I),I=1,M)
   22 CONTINUE
C                           ...
C       .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
C                           ..
      IC=1
      C=1.0
      REWIND 1
      REWIND 2
C                           ...
C     L2 FALSE SIGNALS TAPE MUST BE CREATED FOR USE IN PATERN
C                           ...
      IF (L2) GO TO 52
C                           ...
C       .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
C                           ...
   23 DO 24 I=1,KK
   24 KCOUNT(I)=0
      READ (3) (DATA(I),I=1,M)
      DO 27 I=1,M
      DO 25 K=1,KK
      IF (TRAIN(I).EQ.STD(K)) GO TO 26
   25 CONTINUE
   26 KCOUNT(K)=KCOUNT(K)+1
      J=KCOUNT(K)+KCOUNT(K)
   27 STA(J)=DATA(I)
      J=KCOUNT(KK+1)
      WRITE (1) C,(STA(I),I=1,J),C
      IC=IC+1
      IF (IC.LE.N) GO TO 23
      REWIND 2
      REWIND 1
      REWIND 3
      DO 29 II=1,KK
      DO 28 LL=1,N
      J=KCOUNT(II)+1
      L=KCOUNT(II)
      K=KCOUNT(KK+1)+2-J-L
      READ (1) (SS,I=1,J),(DATA(I),I=1,L),(SS,I=1,K)
   28 WRITE (2) (DATA(I),I=1,L)
   29 REWIND 1
      GO TO 40
C                           ...
C       .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
C     CORRTS INPUT
C                           ...
   30 REWIND 3
      REWIND 9
      DO 31 I=1,N
      LARGE(I)=0
   31 LSL(I)=100000000000
      IF (.NOT.L13) GO TO 32
C      SECOND HALF
      MM1=MX-MX/2
      MM2=MY
      IF(MM1*MM2.EQ.MPQ) GO TO 32
      DO 400 JACK=1,20
      IF(MPQ/IPRIME(JACK)*IPRIME(JACK).NE.MPQ) GO TO 400
      INDEXX=IPRIME(JACK)
      INDEXY=MPQ/IPRIME(JACK)
  400 CONTINUE
      MM1=INDEXX
      MM2=INDEXY
   32 IF (.NOT.L12) GO TO 33
      MPQ=M/2
      IF (MPQ*2.NE.M) MPQ=MPQ+1
      MM1=MX/2
      MM2=MY
      IF(MM1*MM2.EQ.MPQ) GO TO 33
      DO 500 JACK=1,20
      IF(MPQ/IPRIME(JACK)*IPRIME(JACK).NE.MPQ) GO TO 500
      INDEXX=IPRIME(JACK)
      INDEXY=MPQ/IPRIME(JACK)
  500 CONTINUE
      MM1=INDEXX
      MM2=INDEXY
   33 IF (L13) GO TO 34
C                           ...
C     READ IN FORMAT FOR DATA
C                           ...
      READ (F,89) FORM
      CALL CHANGE (FORM,FORM1)
   34 CALL NNAME
      WRITE (6,86)
C                           ...
C       .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
C     OBTAIN DATA AND FIND MAX AND MIN
C                           ...
      IF(.NOT.L13) WRITE(6,100) FORM,FORM1
      DO 38 J=1,N
      IF (.NOT.L13) GO TO 39
      READ (11) TRAIN(J),(DATA(I),I=1,M)
      GO TO 36
   39 READ (F,FORM1) TRAIN(J),(DATA(I),I=1,M)
      IF (.NOT.L12.OR.J/2*2.NE.J) GO TO 36
```

```
C     USE ALL ODD DIMS FOR FIRST HALF
      WRITE (11) TRAIN(J),(DATA(I),I=1,N)
      GO TO 38
   36 WRITE (A,FORMAT) TRAIN(J),(DATA(I),I=1,N)
      IF (.NOT.LF) GO TO 35
C     DATA IS IN INTEGER FORMAT
      DO 37 I=1,N
      IF (DATA(I).LT.LSL(I)) LSL(I)=DATA(I)
   37 IF (DATA(I).GT.LARGE(I)) LARGE(I)=DATA(I)
   38 WRITE (9) (DATA(I),I=1,N)
      IF(.NOT.L17) GO TO 800
      JWRONG=1
      DO 801 I=2,N,2
      JWRONG=JWRONG+1
  801 TRAIN(JWRONG)=TRAIN(I)
  800 CONTINUE
      NUM=
C     CALL FORM IF DATA IS IN INTEGER FORM
      IF(LF)  GO TO 700
      CALL FORT (LSL,LARGE,N,M)
      GO TO 39
  700 IF(.NOT.L15) GO TO 39
C     DATA IS TO ABSOLUTELY QUANTIZED
      LLARGE=LARGE(1)
      LLSL=LSL(1)
      DO 700 J=2,N
      IF(LLARGE.LT.LARGE(J)) LLARGE=LARGE(J)
      IF(LLSL.GT.LSL(J)) LLSL=LSL(J)
  700 CONTINUE
      DO 701 J=1,N
      LARGE(J)=LLARGE
  701 LSL(J)=LLSL
C                          ***
C         *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *
C     QUANTITIZE DATA AND BUILD TRAINING REGION ARRAY WITH KCOUNT   NTS
C                          ***
   39 REWIND 9
      DO 40 J=1,N
   40 ACC(J)=(FLOAT(LEVEL-1))/FLOAT(LARGE(J)-LSL(J))
      DO 42 J=1,M
      READ (9) (DATA(I),I=1,N)
      DO 41 I=1,N
      DATA(I)=FLOAT(DATA(I)-LSL(I))*ACC(I)+.5
   41 CONTINUE
      WRITE (3) (DATA(I),I=1,N)
   42 CONTINUE
      REWIND 2
      STO(1)=TRAIN(1)
      KCOUNT(1)=1
      DO 45 I=2,M
   43 IF (TRAIN(I).EQ.STO(K)) GO TO 44
      K=K+1
      IF (K.LE.KK) GO TO 43
      STO(K)=TRAIN(I)
      KCOUNT(K)=1
      KK=KK+1
      GO TO 45
   44 KCOUNT(K)=KCOUNT(K)+1
   45 K=1
      REWIND 3
      IF (L2) GO TO 52
      DO 47 K=1,KK
      DO 47 L=1,M
      REWIND 3
      II=1
C                          ***
C     STORE      ING REGIONS ON TAPE 3
C                          ***
      DO 46 J=1,M
      READ (3) (STA(I),I=1,N)
      IF (TRAIN(J).NE.STO(K)) GO TO 46
      DATA (II)=STA(L)
      II=II+1
   46 CONTINUE
C                          ***
C     STORE DATA ON TAPE 2
C                          ***
   47 WRITE (2) (DATA(I),I=1,II)
C                          ***
C         *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *
C     QUANTITIZE DATA IN INTEGER LEVELS FROM 0 THROUGH LEVEL.
C     PLACE RESULTS ON TAPE ONE DIMENSION BY DIMENSION
C                          ***
   48 REWIND 3
      REWIND 1
      REWIND 2
      IF (FLOAT(LEVEL).GT.1.E11) STOP
      DO 51 I=1,N
      KK=KCOUNT(I)
      DO 49 J=1,M
   49 STA(J)=0
      DO 50 II=1,N
      READ (2) (STA(J),J=1,KK)
C                          ***
C         *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *
C                          ***
      DO 50 J=1,M
   50 STA(J)=ACC(I)*FLOAT(STA(J)-LSL(I))
   51 WRITE (1) (STA(J),J=1,M)
C                          ***
C     PLACE DATA IN ALTERNATE FORM OF INPUT IF ALT IS SPECIFIED ON
C     THE PARAMETER CARD
C                          ***
   52 IF (L4.AND.(.NOT.L17)) GO TO 55
```

```
      REWIND 2
      REWIND 4
      REWIND 3
      IF (L1) GO TO 72
C                           ***
C     PHOPTS TO CHOPTS SECTION
C                           ***
      NPHO=N
      LOG=.FALSE.
      IC=0
      NT=N
   53 IM=NPHO*M
      IF (.NOT.(IM.GT.10000)) GO TO 54
      NPHO=(NT-((IM-10000)/M+1))
      IM=NPHO*M
      LOG=.TRUE.
   54 DO 55 I=1,NPHO
      J=(I-1)*M+1
      K=I*M
   55 READ (3) (STA(L),L=J,K)
      IF (LOG) GO TO 57
      DO 56 I=1,N
      IF (.NOT.L4) WRITE (43,87) TRAIN(I),(STA(K),K=I,IM,N)
   56 IF (.NOT.L7) WRITE (4) (STA(K),K=I,IM,N)
      GO TO 76
   57 DO 58 I=1,N
   58 WRITE (2) (STA(K),K=I,IM,N)
      IC=IC+1
      NT=NT-NPHO
      KCOUNT(IC)=NPHO
      NPHO=NT
      IF (NT.EQ.0) GO TO 59
      GO TO 53
   59 REWIND 2
      K=10000/N
      L=M-K
      JJJ=0
      DO 61 LL=1,IC
      DO 60 I=1,K
      JJ=JJJ+1
      JJJ=JJ+KCOUNT(LL)-1
   60 READ (2) (STA(II),II=JJ,JJJ)
      DO 61 J=1,L
   61 READ (2) (A,I=JJ,JJJ)
      DO 62 I=1,K
      CALL DEF (ICOUNT,STA,KCOUNT,I,IC,N)
      IF (.NOT.L4) WRITE (43,87) TRAIN(I),(ICOUNT(K1),K1=1,N)
   62 IF (.NOT.L7) WRITE (4) (ICOUNT(K1),K1=1,N)
      REWIND 2
      MULT=1
   63 IF (M.LT.L) GO TO 67
      K1=MULT*K
      JJJ=0
      DO 65 LL=1,IC
      J=KCOUNT(LL)
      DO 64 I=1,K1
   64 READ (2) (A,II=1,J)
      DO 65 J=1,L
      JJ=JJJ+1
      JJJ=JJ+KCOUNT(LL)-1
   65 READ (2) (STA(I),I=JJ,JJJ)
      DO 66 I=1,L
      CALL DEF (ICOUNT,STA,KCOUNT,I,IC,L)
      K=I+K*MULT
      IF (.NOT.L4) WRITE (43,87) TRAIN(K1),(ICOUNT(K1),K1=1,N)
   66 IF (.NOT.L7) WRITE (4) (ICOUNT(K1),K1=1,N)
      GO TO 76
   67 L=L-K
      K1=MULT*K
      JJJ=0
      DO 70 LL=1,IC
      JJ1=KCOUNT(LL)
      DO 68 I=1,K1
   68 READ (2) (A,II=1,JJ1)
      DO 69 J=1,K
      JJ=JJJ+1
      JJJ=JJ+KCOUNT(LL)-1
   69 READ (2) (STA(II),II=JJ,JJJ)
      DO 70 J=1,L
   70 READ (2) (A,I=1,JJ1)
      DO 71 I=1,K
      CALL DEF (ICOUNT,STA,KCOUNT,I,IC,K)
      K1=I+MULT*K
      IF (.NOT.L4) WRITE (43,87) TRAIN(K1),(ICOUNT(K1),K1=1,N)
   71 IF (.NOT.L7) WRITE (4) (ICOUNT(K1),K1=1,N)
      MULT=MULT+1
      REWIND 2
      GO TO 63
C                           ***
C      * * * * * * * * * * * * * * * * * * * * * * * * *
C     CCOPTS TO PHOPTS SECTION
C                           ***
   72 NT=N
      CALL TRANS(TRAIN,M,DATA)
      WRITE (43,88) (TRAIN(I),I=1,N)
      JJJ=0
   73 K=10000/M
      IF (K.GT.NT) K=NT
      DO 74 L=1,K
      READ (3) (ICOUNT(I),I=1,M)
      DO 74 I=1,M
      J=(I-1)*K+L
   74 STA(J)=ICOUNT(I)
      K1=1
```

```
      K2=M                                                        E 431
      JJ=JJJ+1                                                    E 432
      JJJ=JJJ+K                                                   E 433
      DO 75 I=JJ,JJJ                                              E 434
      WRITE (43,89) I                                             E 435
      WRITE(43,99) (STA(J),J=1,42)
   99 FORMAT(13I6)
      K1=K2+1                                                     E 437
   75 K2=K2+M                                                     E 43.
      NT=NT-K                                                     E 439
      IF (NT.GT.0) GO TO 73                                       E 440
C                           ...                                   E 441
C        .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  E 442
C     WRITE OUT TRAINING REGIONS CENTERED ON PAGE                 E 443
C                           ...                                   E 444
   76 IF (L5) GO TO 78                                            E 445
      CALL TRAME (TRAIN,M,DATA)                                   E 446
      WRITE (1) (DATA(I),I=1,M)                                   E 447
      J=1                                                         E 448
      JJ=MM1
      L=(130-MM1)/2
      IF(.NOT.L11) L=(70-MM1)/2
      CALL HNAME                                                  E 452
      WRITE (6,FMT9)                                              E 453
      DO 77 I=1,MM2                                               E 455
      WRITE (6,91) (BLANK,II=1,L),(TRAIN(II),II=J,JJ)             E 455
      J=JJ+1                                                      E 455
   77 JJ=JJ+MM1
      REWIND 3                                                    E 459
   78 IF (L10) WRITE (43,94)                                      E 459
      CALL HNAME                                                  E 460
C                           ...                                   E 461
C     PTINT AND PUNCH QUANTITIZED DATA IF DESIRED                 E 462
C                           ...                                   E 463
      WRITE (6,93)                                                E 464
C      IF L1 IS TRUE THEN CORRTS WAS SPECIFIED
      IF (L1) GO TO 80                                            E 465
      DO 79 I=1,N                                                 E 465
      WRITE(6,777) I
      READ (3) (DATA(J),J=1,M)                                    E 467
      IF (L10) WRITE (43,95) (DATA(J),J=1,M)                      E 469
   79 WRITE (6,92) (DATA(J),J=1,M)                                E 469
      GO TO 82                                                    E 470
   80 DO 81 I=1,N                                                 E 471
      READ (3) (DATA(J),J=1, )                                    E 472
      IF (L10) WRITE (43,95) (DATA( ),J=1,N)                      E 473
   81 WRITE (6,92) (DAT   ),  =1, )                               E 474
   82 IF (.NOT.LA.AND.F     ) GO TO 83                            E 475
      KR=MAX0(KR,KK)                                              E 476
      REWIND 1                                                    E 477
      CALL CHNET                                                  E 478
   83 WRITE (6,98)                                                E 479
      WRITE (6,96)                                                E 480
      WRITE (6,98)                                                E 481
      WRITE (6,97) KW,KR                                          E 482
      WRITE (6,98)                                                E 483
      WRITE (6,96)                                                E 484
      WRITE (6,98)                                                E 485
      STOP                                                        E 486
C                           ...                                   E 487
C                                                                 E 488
                            ...                                   E 489
   84 FORMAT (15HALTERNATE PUNCH)                                 E 490
   85 FORMAT (12A6)                                               E 491
   86 FORMAT (1X,13HORIGINAL DATA/)                               E 492
   87 FORMAT (A6,3X,10(1X,I10))                                   E 493
   88 FORMAT (80A1)                                               E 494
   89 FORMAT (5HPHOTO,24,I3)                                      E 495
   90 FORMAT (15I6)                                               E 496
   91 FORMAT (1X,131A1)                                           E 497
   92 FORMAT (10(1X,I10))                                         E 498
   93 FORMAT (1X,14HQUANTIZED DATA/)                              E 499
   94 FORMAT (27HQUANTIZED DATA FORMAT(8I10))                     E 500
   95 FORMAT (8I10)                                               E 501
   96 FORMAT (103H ...................................................  E 502
     1...................................                    )    E 503
   97 FORMAT (46H ONE OF US CAN NOT COUNT  I FIND THAT THERE ARE ,I3,27H  E 504
     1TRAINING REGIONS  YOU FOUND,I3)                             E 505
   98 FORMAT (//)                                                 E 506
      END                                                         E 507
```

17584 01 05-02-69   10.247

```
      SUBROUTINE FPR (LSL,LARGE,M,N)
      DIMENSION SS(20), BB(20), LSL(20), LARGE(20), IDATA(13)
      COMMON PPO1,FORMT1(12)
      COMMON I
      COMMON FMT1(15),FMT2(3),FMT9(5),FMT11(6),FMT12(7),FMT13(3),FMT14(2
     1),FMT16(2),FMT22(3),FMT56(6),FMT57(4),FMT58(8)
      COMMON L8,L9,L10,L11,L12,L13,L14
      COMMON K,L,B(8879),DATA(8000)
      EQUIVALENCE (IDATA(1),DATA(1))
      INTEGER F
      LOGICAL L8,L9,L10,L11,L12,L13,L14
      REWIND 9
      REWIND 21
C                        ***
C      L14 IS TRUE IF ABSOLUTE QUANTIZATION
C      PHOTO FORM
C      N IS THE NUMBER OF PHOTOGRAPHS
C      M IS THE NUMBER OF DATA POINTS
C     READ IN DATA
C                        ***
      S=1.E20
      B=-1.E20
C      GO THRU EACH PHOTOGRAPH
      DO 3 I=1,N
      READ (9) (DATA(J),J=1,M)
C                        ***
C     SEARCH FOR MAX AND MIN VALUES IN DATA
C                        ***
      DO 1 J=1,M
      IF (S.GT.DATA(J)) S=DATA(J)
    1 IF (B.LT.DATA(J)) B=DATA(J)
      IF (L14) GO TO 3
      R=1./(B-S)*1.E10
C                        ***
C     QUANTITIZE AND STORE DATA ON 21
C      QUANTIZE RELATIVELY
C                        ***
      DO 2 J=1,M
    2 IDATA(J)=(DATA(J)-S)*R
      S=1.E20
      B=-1.E20
    3 WRITE (21) (IDATA(J),J=1,M)
      REWIND 9
      REWIND 21
C                        ***
C     STORE DATA BACK ON 9
C      FOR EACH PHOTOGRAPH
C                        ***
      DO 5 I=1,N
      READ (21) (IDATA(J),J=1,M)
      IF (.NOT.L14) GO TO 5
C      QUANTIZE ABSOLUTELY
      R=1./(B-S)*1.E10
      DO 4 J=1,M
    4 IDATA(J)=(DATA(J)-S)*R
    5 WRITE (9) (IDATA(J),J=1,M)
    6 DO 7 I=1,10
      LSL(I)=0
    7 LARGE(I)=10000000000
      RETURN
      ENTRY FPR1(LSL,LARGE,M,N)
      REWIND 9
      REWIND 21
C                        ***
C      CORPTS FORM
C      M IS THE NUMBER OF PHOTOGRAPHS
C      N IS THE NUMBER OF DATA POINTS
C      L14 IS TRUE FOR ABSOLUTE QUANTIZATION
C     INITIALIZE
C                        ***
      DO 8 I=1,M
      SS(I)=1.E20
    8 BB(I)=-1.E20
C                        ***
C     SEARCH FOR EXTREME VALUES
C                        ***
C      GO THRU EACH POINT
      DO 9 I=1,N
C      READ IN ALL THE COMPONENTS FOR THAT POINT
      READ (9) (DATA(J),J=1,M)
      WRITE (21) (DATA(J),J=1,M)
      DO 9 J=1,M
      IF (SS(J).GT.DATA(J)) SS(J)=DATA(J)
    9 IF (BB(J).LT.DATA(J)) BB(J)=DATA(J)
      IF (.NOT.L14) GO TO 12
      S=1.E20
      B=-1.E20
      DO 10 I=1,M
      IF (S.GT.SS(I)) S=SS(I)
   10 IF (B.LT.BB(I)) B=BB(I)
      DO 11 I=1,M
      SS(I)=S
   11 BB(I)=B
   12 REWIND 9
      REWIND 21
C                        ***
C      QUANTIZE
C                        ***
      DO 13 I=1,M
   13 BB(I)=1./(BB(I)-SS(I))*1.E10
      DO 15 I=1,N
      READ (21) (DATA(J),J=1,M)
C                        ***
C     STORE DATA ON 9
C                        ***
      DO 14 J=1,M
   14 IDATA(J)=(DATA(J)-SS(J))*BB(J)
   15 WRITE (9) (IDATA(J),J=1,M)
      GO TO 6
      END
```

<><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
*   23005 WORDS OF MEMORY USED BY THIS COMPILATION
<><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>

```
17964 01  05-02-69   10.269

      SUBROUTINE DEF (L,M,N,I,IC,K)
      DIMENSION L(2), M(2), N(2)
      JT=(I-1)*N(1)
C                    ***
C     SET SINGLE CHARACTER TRAINING REGION
C     DEFINE IN ALPHABETICAL ORDER
C                    ***
      JS=0
      KC=0
      DO 2 J=1,IC
      JJ=M(J)
      DO 1 II=1,JJ
      JT=JT+1
      JS=JS+1
    1 L(JS)=M(JT)
      KC=KC+JJ+N
    2 JT=(I-1)*N(J+1)
      RETURN
      END

<><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
*    23015 WORDS OF MEMORY USED BY THIS COMPILATION
<><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
```

```
17964 01  05-02-69   10.269

      SUBROUTINE TRANS (A,M,DATA)
      COMMON NPCL,FORMT1(12)
      COMMON F
      COMMON FMT1(15),FMT2(3),FMT9(5),FMT11(6),FMT12(7),FMT13(3),FMT14(8
     1),FMT16(2),FMT22(3),FMT56(6),FMT57(6),FMT58(8)
      DIMENSION A(2), B(36), D(36)
      INTEGER DATA
      DIMENSION DATA(2)
      INTEGER F
      DATA D/1HA,1HB,1HC,1HD,1HE,1HF,1HG,1HH,1HI,1HJ,1HK,1HL,1HM,1HN,1HO
     1,1HP,1HQ,1HR,1HS,1HT,1HU,1HV,1HW,1HX,1HY,1HZ,1H1,1H2,1H3,1H4,1H5,1
     2H6,1H7,1H8,1H9,1H0/
C                    ***
C     CREATE CROSS REFERENCE TABLE FOR TRAINING REGIONS
C                    ***
      CALL MNAME
      WRITE (6,FMT1) D(1),A(1)
      B(1)=A(1)
      A(1)=D(1)
      DATA (1)=1
      KK=1
      K=1
      DO 3 I=2,M
    1 IF (A(I).EQ.B(K)) GO TO 2
      K=K+1
      IF (K.LE.KK) GO TO 1
      WRITE (6,FMT2) D(K),A(I)
      B(K)=A(I)
      A(I)=D(K)
      DATA (I)=K
      KK=KK+1
      GO TO 3
    2 A(I)=D(K)
      DATA (I)=K
    3 K=1
      RETURN
      END

<><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
*    23017 WORDS OF MEMORY USED BY THIS COMPILATION
<><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
```

```
17964 02  05-02-69   10.276

                                         1            SYMREF  CHANGE
   000000  000014 2350 00  010           2 CHANGE     LDA     T1
   000001  000020 7550 00  010           3            STA     T2
   000002  000002 0200 31  000           4            EAXO    2,1*
   000003  000022 7400 00  010           5            STXO    T2
   000004  000017 2350 00  010           6            LDA     T3
   000005  000021 7550 00  010           7            STA     LOC
   000006  000003 6200 31  000           8            EAX0    3,1*
   000007  000021 7400 00  010           9            STX0    LOC
   000010  000022 2350 00  010          10            LDA     OCT
   000011  000003 7550 31  000          11            STA     3,1*
   000012  000020 2350 52  010          12 J.         LDA     T2,DC
   000013  000021 7550 52  010          13            STA     LOC,DC
   000014  000012 6070 00  010          14            TTF     J.
   000015  000010 7100 11  000          15            TRA     0,1
   000016  000004 0100 01  000          16 T1         TALLY   **,60,1
   000017  000004 0100 04  000          17 T2         TALLY   **,60,4
                        000020          18 T3         BSS     1
                        000021          19 LOC        BSS     1
   000022  000010772220          000    20 OCT        OCT     1,1,1.

     ERROR LINKAGE

   000023  000000000000          000
 END OF BINARY CARD 00000002
   000024  233021452725          000
 END OF BINARY CARD 00000003
                                        21            END
    26 IS THE NEXT AVAILABLE LOCATION, GMAP AID 000000
 THERE WERE     NO  WARNING FLAGS IN THE ABOVE ASSEMBLY
```

17564 03 05-02-69 10.284

```
      SUBROUTINE PATERN
      COMMON MPD1,FORMT1(12)
      COMMON F
      COMMON FMT1(15),FMT2(3),FMT9(5),FMT11(6),FMT12(7),FMT13(3),FMT14(2
     1),FMT1A(2),FMT22(3),FMT56(6),FMT57(4),FMT58(8)
      COMMON L8,L9,L10,L11,L12,L13,L14
      COMMON NP,MM,LEVEL,NZ1,NPOINT,ITAPE,AAA,LT,L1,L2,L3,L4,L5,L6,N,M1,
     1M2,NDUMY,L7,MD,ME,KC,ZZZ
C                          ***
C     N=NUMBER OF TRAINING REGIONS
C     LEVEL=NUMBER OF LEVELS OF QUANTITIZATION
C     NP=NUMBER OF PHOTOS
C     NZ1=NUMBER OF SIGNIFICENT FIGURES PER DATA POINT
C     NPOINT=ORDERED LIST OF POINTS PER TRAINING REGION
C      INPUT IS EXPECTED ON TAPE UNIT *ONE*
C     LT=LOGICAL TRUE FOR NORMALIZED AND FALSE FOR UNORMALIZED
C                          ***
      DIMENSION AAA(625)
      DIMENSION ZZZ(24000), IPT(2), IREM(2), NREG(2), NPOINT(36), KPT(36
     1), RESULT(625), SIGMA(36), FORM(12), BLANK(4)
      EQUIVALENCE (ZZZ(1),IPT(1)), (ZZZ(1),IREM(1)), (ZZZ(1),NREG(1))
      LOGICAL LT,L1,L2,L3,L4,L5,L6,L7
      LOGICAL L8,L9,L10,L11,L12,L13
      INTEGER F
      DATA BLANK/6HUNORMA,5HLIZED,6HNORMAL,4HIZED/
      REWIND 1
      REWIND 2
      REWIND 9
      REWIND 10
      REWIND 20
C                          ***
C     PRINT HEADING FOR PUNCH
C     NHIGH IS ABSOLUTE MAX NUMBER OF POINTS IN A SINGLE TRAINING REGION
C                          ***
      WRITE (10) N
      WRITE (43,32)
      NHIGH=0
      NTOTAL=0
C                          ***
C     NPOINT CONTAINS NO. OF POINTS PER TRAINING REGION
C     COUNT TOTAL PTS.     FIND MAX IN NPOINT
C                          ***
      DO 1 I=1,N
      NTOTAL=NTOTAL+NPOINT(I)
      IF (NPOINT(I).LT.NHIGH) GO TO 1
      NHIGH=NPOINT(I)
    1 CONTINUE
      J=1
      IF (LT) J=3
      K=J+1
      CALL NNAME
      WRITE (6,33) (BLANK(I),I=J,K)
      WRITE (6,34)
C                          ***
C     PRINT PARAMETERS
C                          ***
      WRITE (6,35) N,LEVEL,NP
C                          ***
C     INITIALIZE
C                          ***
      DO 2 I=1,102
    2 RESULT(I)=0.
C                          ***
C     RESULT IS PROBABILITY MATRIX
C     CALCULATE NO. OF LOCATIONS FOR USE IN IREM AND IPT
C     NEVER MORE THAN 1 TR IN BOTTOM OF ZZZ
C                          ***
      NG=20000-NHIGH
      IF (NG.LT.10000) NG=10000
      NBC=0
      NC=0
      IP=0
      SP=0.
C                          ***
C     CALCULATE NORMALIZING FACTORS
C                          ***
      DO 3 I=1,N
      SIGMA(I)=1./FLOAT(NPOINT(I))
    3 SP=SP+FLOAT(NPOINT(I))
C                          ***
C     KCOUNT IS A FUNCTION OF THE POSITION IN THE TOP PART OF ZZZ
C     INDICATION WHERE THE PRESENT UNIQUE N-TUPLE(UNIQUE TO TR)
C     WILL BE STORED
C     LICN IS THE BEGINNING OF THE BLOCK KI IN THE TOP HALF OF ZZZ
C     IPOI IS NUMBER OF POINTS IN KI TRAINING REGION
C     AND LOAD IT UP IN THE BOTTOM PART OF ZZZ
C     STARTING AT THE TOP AND COMMING DOWN
C                          ***
      KCOUNT=1
      LICN=1
C                          ***
C     KI RANGES OVER NO. OF TRAINING REGIONS
C                          ***
      DO 8 KI=1,N
      IPOI=NPOINT(KI)
C                          ***
C     READ IN SINGLE TRAINING REGION
C                          ***
      READ (1) (NREG(I),I=1,IPOI)
C                          ***
C     SET N-TUPLE IN EVEN LOCATIONS
C     PUT NO. TIMES OCCURED IN ODD LOCATIONS
C     TRAINING REGION IS STORED IN LOWER HALF OF ODD LOCATIONS
```

17564 03 05-02-69 10.284

```
C                              ***
      DO 7 LJ=1,IPOI
      KKKK=NREG(LJ)
      IF (LICK.EQ.NCOUNT) GO TO 5
      II=NCOUNT-1
      DO 4 I=LICK,II
      I1=20002-I*2
      IF (KKKK.EQ.IREM(I1)) GO TO 6
    4 CONTINUE
C                              ***
C     IF OVERLAP OCCURES IN SCRATCH AREA GO TO 777 AND STORE RESULTS
C     NREG(LJ) HAS NOT OCCURED BEFOR  IN THE TRAINING REGION, MAKE
C     POSITION TO STORE IT
C                              ***
    5 IF (NCOUNT*2.GT.NB) GO TO 27
      NCOUNT2=20002-NCOUNT*2
      IREM(NCOUNT2)=KKKK
      IPT(NCOUNT-1)=10000+K*100
      NCOUNT=NCOUNT+1
      GO TO 7
C                              ***
C     NREG(LJ) HAS OCCURED BEFOR IN TO KI
C                              ***
    6 I1=20001-I*2
      IPT(I1)=IPT(I1)+10000
    7 CONTINUE
      LICK=NCOUNT
    8 NN=NCOUNT-1
C                              ***
C     N2 IS THE LOWEST LOCATION ON THE TOP PART WHICH IS USED
C                              ***
      I=2*NN
      N2=20001-I
C                              ***
C     STORE RESULTS ON TAPE 10
C                              ***
      WRITE (10) I
      WRITE (10) (IREM(I),I=N2,20000)
    9 IF (NRC.EQ.0) GO TO 11
      JTAPE=2
      LTAPE=3
      REWIND 2
      REWIND 3
      LL=1
   10 READ (JTAPE) NCOUNT
      N=20001-NCOUNT
      NM1=N2-1
      KK=N2-4
      READ (JTAPE) (IREM(I),I=N,NM1)
      NN=KK
   11 L=NN-1
C                              ***
C     SEQUENCE N-TUPLES IN DECENDING ORDER
C                              ***
      DO 12 I=1,L
      IP1=I+1
      DO 12 J=IP1,NN
      I1=20002-I*2
      J1=20002-J*2
      IF (IREM(I1).GE.IREM(J1)) GO TO 12
      IPOI=IREM(I1)
      IREM(I1)=IREM(J1)
      IREM(J1)=IPOI
      IPOI=IPT(I1-1)
      IPT(I1-1)=IPT(J1-1)
      IPT(J1-1)=IPOI
   12 CONTINUE
      IF (NRC.EQ.0) GO TO 15
      WRITE (LTAPE) (IREM(I),I=NN,NM1)
      LL=LL+1
      IF (LL.LE.NRC) GO TO 10
      KP1=KKK+2
      KKK=IREM(KKK+1)
      DO 13 I=KP1,20000,2
      IF (IREM(I).NE.KKK) GO TO 14
   13 CONTINUE
   14 NN=I/2
   15 CONTINUE
      ICOUNT=0
      I=1
C                              ***
C     LOOK FOR STARTING AND END POINTS OF EACH N-TUPLE
C     LOOK FROM PRESENT N-TUPLE DOWN
C                              ***
   16 DO 17 IJ=I,NN
      J=IJ
      I1=20002-I*2
      J1=20002-J*2
      IF (IREM(I1).NE.IREM(J1)) GO TO 18
   17 CONTINUE
      J=NN+1
   18 J=J-1
C                              ***
C     J IS A FUNCTION OF THE LAST N-TUPLE LIKE THE I TH N-TUPLE TOP OFF
C                              ***
      K =1
      IF (I.EQ.J) GO TO 19
C                              ***
C     N-TUPLE OCCURS IF MORE THAN ONE TO
C                              ***
      GO TO 20
C                              ***
C     IF ONLY ONE T.R. IN N-TUPLE CLASSIFY N-TUPLE
C                              ***
```

17564 03 05-02-69 10.284

```
   19 I1=20001-I+2
C                         ***
C     I1 IS THE LOCATION OF THE CONT FOR THE I TH N-TUPLE
C                         ***
      N=IPT(I1)/10000
      N=(IPT(I1)-N*10000)/100
C                         ***
C     N IS NUMBER OF TIMES THAT THE N-TUPLE IN I1+1 OCCURED
C                         ***
      GO TO 21
C                         ***
C     IF TWO OR MORE T.O. IN N-TUPLE LOOK FOR PREDOMINANT T.O.
C     AND CLASSIFY
C                         ***
   20 CALL DECIDE (I1,J,IPT,SIGMA,K,N,AAA,LT,NTOTAL,SPROB)
   21 DO 22 L=1,J
      K1=20001-L+2
   22 IPT(K1)=IPT(K1)+0
      NCOUNT=NCOUNT+1
C                         ***
C     STORE IN NREG N-TUPLE AND CLASSIFICATION
C     IF NREG FULL GO TO 291 AND STORE NREG
C                         ***
   23 IF (NCOUNT.GT.NHIGH) GO TO 30
      I1=20002-I+2
      NREG(NCOUNT)=IPT(I1)
      WRITE (9) K,SPROB
      I=J+1
      IF (I.LE.NN) GO TO 19
      NCOUNT=NCOUNT+1
      IF (NI.NE.0) GO TO 30
C                         ***
C     CLASSIFY ALL N-TUPLES
C     NK IS TO FOR THE N-TUPLE
C     NKK IS THE CLASSIFICATION BY BAYES' RULE
C     LJ IS POSITION IN RESULT ARRAY FOR (NK,NKK)
C                         ***
      DO 24 I=1,NN
      I1=20001-I+2
      N=IPT(I1)/10000
      NK=(IPT(I1)-N*10000)/100
      NKK=IPT(I1)-N*10000-NK*100
      LJ=(NK-1)*N+NKK
C                         ***
C     IF DATA WAS STORED GO FETCH IT
C                         ***
   24 RESULT(LJ)=RESULT(LJ)+FLOAT(N)
      IF (NOC.GT.0) GO TO 30
      SUM=0.
C                         ***
C     CALCULATE PROBABILITIES AND PRINT OUT RESULTS
C                         ***
      NP1=N+1
      NN1=N*N
      DO 25 I=1,NN1,NP1
   25 SUM=SUM+RESULT(I)
C                         ***
C     SUM IS TOTAL PROB OF CORRECT CLASSIFICATIONS TO CONDITIONAL PROB
C                         ***
      SUM=SUM/SP
      DO 26 I=1,N
      LJ=(I-1)*N+1
      K=I+N
      DO 26 J=LJ,K
   26 RESULT(J)=RESULT(J)*SIGMA(I)
      CALL OUTP (RESULT,N)
      WRITE (6,FMT501) SUM
C                         ***
C     WRITE END OF FILE MARKS ON TAPES
C                         ***
      I=0
      WRITE (10) I
      WRITE (20) I
      REWIND 10
      CALL CHAT
C                         ***
C     STORE RESULTS AND INITIALIZE NCOUNT
C     NCOUNT IS NOW THE NUMBER OF LOCATIONS ACTUALLY FILLED
C     NCOUNT IS THE BOTTOM MOST POSITION FILLED IN THE TOP PART OF III
C     DUMP TOP PATH OF III ONTO 2
C     NOC GIVES NUMBER OF DUMPS ON 2
C                         ***
   27 NCOUNT=(NCOUNT-1)+2
      NCOUNT=20001-NCOUNT
      WRITE (2) NCOUNT
      WRITE (2) (IPT(I1),IPT(I1+1),I1=NCOUNT,20000,2)
      NCOUNT=1
      LICK=1
      NOC=NOC+1
      GO TO 9
   28 L=20001-NN+2
      KK=NN-1
      I=20001-L+NN
      IP1=I+1
      DO 29 J=KK,L
      I=I+1
   29 IPT(I)=IPT(J)
      JAL=TAPE
      LTAPE=JTAPE
      JTAPE=J
      NOC=NOC-1
      NN=L-NN+1
      GO TO 9
C                         ***
C     STORE, PUNCH AND INITIALIZE N-TUPLES
C                         ***
   30 NCOUNT=NCOUNT-1
      WRITE (20) NCOUNT
      REWIND 9
      WRITE (20) (NREG(I),I=1,NCOUNT)
      DO 31 I=1,NCOUNT
      READ (9) K1,SPROB
      WRITE (20) NREG(I),K1,SPROB
   31 NREG(I)=0
      WRITE (20) (NREG(I),I=1,NCOUNT)
      REWIND 9
      K=1
      NCOUNT=1
      NCOUNT=1
      GO TO 23
C                         ***
C
C                         ***
   50 FORMAT (22H0TOTAL CLASSIFICATIONS,F10.5 )
   51 FORMAT (1H0,20X/)
   52 FORMAT (1H1,20X, TRAINING PATTERNS,I6,5X, LEVELS ,I5,5X, 0
     1H0TES)
   53 FORMAT (I10,3X,I5,I5)
   54 FORMAT (I10,I10,F10.5)
      END
```

****
*   22110 WORDS OF MEMORY USED BY THIS COMPILATION
****

```
      SUBROUTINE OUTP (RESULT,N)
      DIMENSION RESULT(N,N), IDL(23)
      COMMON MP31,FORMT1(12)
      COMMON F
      COMMON FMT1(15),FMT2(3),FMT8(5),FMT11(6),FMT12(7),FMT13(3),FMT14(2
     1),FMT16(2),FMT22(3),FMT56(6),FMT57(4),FMT58(8)
      COMMON L8,L9,L10,L11,L12,L13,L14
      INTEGER F
      DATA IDL/1HT,1HR,1HU,1HE,1H ,1HT,1HR,1HA,1HI,1HN,1HI,1HN,1HG,1H ,1
     1HR,1HE,1HG,1HI,1HO,1HN/
C                        ***
C     PRINT OUT PROBABILITY MATRIX IN PUBLISHABLE FASHION
C                        ***
      II=1
      III=8
      NN=N
    1 CALL NNAME
      K=NN
      IF (K.LT.8) GO TO 9
      LINE=0
      K=8
      NS=25-N
      IF (NS.EQ.0) GO TO 3
      DO 2 I=1,NS
      LINE=LINE+1
    2 WRITE (6,15)
    3 LINE=LINE+7
      WRITE (6,FMT56)
      WRITE (6,FMT12)
      WRITE (6,FMT13) (I,I=II,III)
      IF (LINE.NE.24) GO TO 4
      WRITE (6,FMT14) IDL(1)
    4 DO 6 J=1,N
      LINE=LINE+1
      WRITE (6,FMT57) J,(RESULT(I,J),I=II,I
      IF (44-LINE.LE.0) GO TO 5
      L=LINE-23
      IF (L.LE.0) GO TO 5
      WRITE (6,FMT14) IDL(L)
    5 LINE=LINE+1
      WRITE (6,15)
      IF (44-LINE.LT.0) GO TO 6
      L=LINE-23
      IF (L.LE.0) GO TO 6
      WRITE (6,FMT14) IDL(L)
    6 CONTINUE
    7 LINE=LINE+1
      L=LINE-23
      IF (L.GT.23) GO TO 8
      WRITE (6,FMT16) IDL(L)
      GO TO 7
    8 NN=NN-K
      II=II+8
      III=III+8
      IF (NN.GT.0) GO TO 1
      RETURN
    9 DO 10 I=1,23
   10 WRITE (6,15)
      WRITE (6,FMT56)
      WRITE (6,FMT12)
      III=II+K-1
      WRITE (6,FMT13) (I,I=II,III)
      WRITE (6,FMT16) IDL(1)
      LINE=24
      DO 12 J=1,N
      LINE=LINE+1
      WRITE (6,FMT57) J,(RESULT(I,J),I=II,III)
      IF (44-LINE.LE.0) GO TO 11
      L=LINE-23
      IF (L.LE.0) GO TO 11
      WRITE (6,FMT14) IDL(L)
   11 LINE=LINE+1
      WRITE (6,15)
      IF (44-LINE.LE.0) GO TO 12
      L=LINE-23
      IF (L.LE.0) GO TO 12
      WRITE (6,FMT14) IDL(L)
   12 CONTINUE
   13 LINE=LINE+1
      L=LINE-23
      IF (L.GT.23) GO TO 14
      WRITE (6,FMT16) IDL(L)
      GO TO 13
   14 RETURN
C                        ***
C
C                        ***
   15 FORMAT (/)
      END
```

```
<><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
*    22939 WORDS OF MEMORY USED BY THIS COMPILATION
<><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
```

17564 03 05-02-69 18.316

```
      SUBROUTINE DECSON (I,J,M1,SIGMA,K,N,A,LT,NTOTAL,XPROB)
      DIMENSION M1(2), SIGMA(2)
      DIMENSION A(2), C(30)
      LOGICAL LT
      M=0
      DO 1 K=1,N
    1 C(K)=0.
      XPROB=0.
      ZLARGE=-.1E+21
      DO 2 K1=I,J
      K=20001-2*K1
      LL=M1(K)/10000
      L=(M1(K)-LL*10000)/100
    2 C(L)=C(L)+FLOAT(LL)
      XPROB=FLOAT(M)/FLOAT(NTOTAL)
      M=1
C                              ***
C     IF NORMALIZED IS SPECIFIED MULT BY NORMALIZING FACTORS
C                              ***
      IF (.NOT.LT) GO TO 4
      DO 3 L=1,N
      IF (C(L).LT.1.E-10) C(L)=0.
    3 C(L)=C(L)*SIGMA(L)
    4 DO 6 L=1,N
      SMALL=0.
C                              ***
C     A IS THE LOSS MATRIX
C                              ***
      DO 5 K=1,N
      SMALL=SMALL+A(M)*C(K)
    5 M=M+1
      SMALL=SMALL-10000.
C                              ***
C     LOOK FOR PREDONOMINT [illegible] REGION
C                              ***
      IF (SMALL.GT.ZLA[illegible]
      ZLARGE=SMALL
      K=L
    6 CONTINUE
C                              ***
C     K IS THE RESULTING TRAINING REGION
C                              ***
      RETURN
      END
```

<><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
*   22985 WORDS OF MEMORY USED BY THIS COMPILATION
<><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>

```
      SUBROUTINE DMP
      DIMENSION IA(20), IB(20), IC(20)
      COMMON MPO1,FORMT1(12)
      COMMON F
      COMMON IDUMY(752),IZZ(24000)
      EQUIVALENCE (IA(1),IZZ(1)), (IB(1),IZZ(12001))
      INTEGER F
      REWIND 9
      REWIND 2
      REWIND 10
      ITAPE=10
      KTAPE=9
      READ (10) N
      IF (N.GT.20) GO TO 19
    1 READ (ITAPE) I
      IF (I.GT.12000) GO TO 19
      IF (I.LE.0) GO TO 6
      ITEST=I
      READ (ITAPE) (IZZ(J),J=1,I)
    2 READ (ITAPE) K
      IMP1=I+1
      L=I+K
      IF (K.LE.0) GO TO 3
      READ (ITAPE) (IZZ(J),J=IMP1,L)
    3 LM1=L-1
      DO 4 II=2,LM1,2
      IP1=II+2
      DO 4 JJ=IP1,L,2
      IF (IZZ(II).GT.IZZ(JJ)) GO TO 4
      IS=IZZ(II-1)
      IZZ(II-1)=IZZ(JJ-1)
      IZZ(JJ-1)=IS
      IS=IZZ(II)
      IZZ(II)=IZZ(JJ)
      IZZ(JJ)=IS
    4 CONTINUE
      WRITE (KTAPE) K
      IF (K.LE.0) GO TO 5
      WRITE (KTAPE) (IZZ(J),J=IMP1,L)
      GO TO 2
    5 WRITE (2) I
      WRITE (2) (IZZ(J),J=1,I)
      KK=KTAPE
      KTAPE=ITAPE
      ITAPE=KK
      REWIND 9
      REWIND 10
      GO TO 1
    6 M1=0
      WRITE (2) M1
      REWIND 2
      WRITE (6,16)
      WRITE (6,17) (I,I=1,N)
    7 DO 8 I=1,20
    8 IC(I)=0
      READ (2) M
      IF (M.LE.0) GO TO 12
      READ (2) (IZZ(I),I=1,M)
      IMEM=IZZ(2)
      M1=IZZ(1)/10000
      M2=(IZZ(1)-10000*M1)/100
      IC(M2)=M1
      DO 11 I=3,M,2
      IF (IMEM.NE.IZZ(I+1)) GO TO 9
      M1=IZZ(I)/10000
      M2=(IZZ(I)-10000*M1)/100
      IC(M2)=M1+IC(M2)
      GO TO 11
    9 WRITE (6,18) IMEM,(IC(J),J=1,N)
      IMEM=IZZ(I+1)
      DO 10 J=1,20
   10 IC(J)=0
      M1=IZZ(I)/10000
      M2=(IZZ(I)-10000*M1)/100
      IC(M2)=M1
   11 CONTINUE
      WRITE (6,18) IMEM,(IC(J),J=1,N)
      GO TO 7
   12 NN=IDUMY(742)
      REWIND 20
   13 READ (20) N
      IF (N.LE.0) GO TO 14
      READ (20) (IA(J),J=1,N)
      READ (20) (IB(J),J=1,N)
      WRITE (6,19) (IA(J),IB(J),J=1,N)
      GO TO 13
   14 CALL CHKPT
   15 WRITE (6,20)
      CALL CHKPT
      RETURN
C                         ***
C
C                         ***
   16 FORMAT (1H1,10,8HN-TUPLES,40X,19HTRAINING REGIONS//)
   17 FORMAT (/,8X,20I5)
   18 FORMAT (5X,I10,5X,20I5,(40X)/)
   19 FORMAT(5(I8,I10,10X,I2))
   20 FORMAT (78H TOO MANY TRAINING REGIONS FOR CONVIENT OUTPUT EXECUTIO
     1N CONTINUED     )
      END
```

<>><>><>><>><>><>><>><>><>><>><>><>><>><>><>><>><>><>><>><>><>><>><>><>><>><>><>><>>

*   22900 WORDS OF MEMORY USED BY THIS COMPILATION

17904 03 09-02-69 10.936

```
      SUBROUTINE PLOTER
      DIMENSION CPS(20,130), PLOT(130), DATA(30), NPT(1000), KLAS(1000),
     1 ENT(1000)
      DIMENSION PLT(130)
      DIMENSION AAA(25), NPOINT(30)
      COMMON /IST/ IST
      COMMON NPD1,FORMT(12)
      COMMON F
      COMMON FMT1(15),FMT2(3),FMT9(5),FMT11(8),FMT12(7),FMT13(3),FMT14(2
     1),FMT15(2),FMT22(3),FMT56(8),FMT57(8),FMT58(8)
      COMMON L8,L9,L10,L11,L12,L13,L14
      COMMON NP,NN,LEVEL,NZ1,NPRINT,ITAPE,AAA,LT,L1,L2,L3,L4,L5,L6,NA,NB
     1,NC,ND,L7,NS,NY,ITEM,ZOZ(24000)
      EQUIVALENCE (KLAS(1),ENT(1)),(ZOZ(1),NPT(1)),(ZOZ(1000)),KLAS(1))
     1 , (ZOZ(2300)),CPS(1,1))
      LOGICAL LT,L1,L2,L3,L4,L5,L6,L7
      LOGICAL L8,L9,L10,L11,L12,L13
      INTEGER F
      INTEGER CPS
      INTEGER PLT
      DATA BLANK/1H /
C      NP IS NUMBER OF PHOTOS
C      NN IS NUMBER OD DATA POINTS PER PHOTO
C      PLOT ARRAY OUTPUTS PHOTO CLASSIFICATION LINE BY LINE
C      NO MORE THAN 20 PHOTOS CAN BE HANDLED HERE
      REWIND 3
      IST=35435900793
      REWIND 4
      REWIND 20
      IF (L9) WRITE (43,12)
      CALL NNAME
      J1=1
      J2=0
    1 READ (20) NCOUNT
      IF (NCOUNT.LE.0) GO TO 2
      J2=J2+NCOUNT
C      NPT IS THE COMPACTED POINT ARRAY
C      KLAS IS THE CATEGORY WHICH NPT IS PUT IN
      READ (20) (NPT(J),J=J1,J2)
      READ (20) (KLAS(J),J=J1,J2)
      J1=J2+1
      GO TO 1
    2 WRITE (6,FMT11)
      J3=(130-NS)/2+1
      J4=J3+70
      REWIND 2
      DO 7 I=1,NY
C      READ A LINE OF UNCOMPACTED N-TUPLES
      DO 3 J=1,NS
      IF (L1) READ (3) (CPS(K,J),K=1,NP)
    3 IF (.NOT.L1) READ (4) (CPS(K,J),K=1,NP)
      DO 4 J=1,130
    4 PLOT(J)=BLANK
      JJ=(130-NS)/2
      DO 6 J=1,NS
C      FOR EACH N-TUPLE IN LINE COMPACT IT AND FIND OUT HOW TO CLASSIFY
      JJ=JJ+1
      KK=0
      DO 5 K=1,NP
    5 KK=KK*LEVEL+CPS(K,J)
      CALL SEARCH (NPT,KLAS,KK,SYMBOL,J2,NP,N,LEVEL)
C      PLT CONTAINS NUMERIC CODE FOR SYMBOL
      PLT(J)=K
    6 PLOT(JJ)=SYMBOL
      WRITE (2) (PLT(J),J=1,NS)
C      NOW WRITE OUT THE LINE
      IF (L9) WRITE (43,13) (PLOT(J),J=J3,J4)
      IF (.NOT.L11) WRITE (6,14) (PLOT(J),J=31,100)
    7 IF (L11) WRITE (6,14) (PLOT(J),J=1,130)
      ITEM=0
      IF (L2.AND.L5) GO TO 11
C      COMPUTE CONTINGENCY TABLE HERE
      REWIND 1
      REWIND 2
      DO 8 I=1,NA
      NC=NPOINT(I)
    8 READ (1) (ZOZ(J),J=1,NC)
C      READ IN TRAINING REGION
      NS=NA*NA
      N=NS+1
      NPN=N+NN*NS
      READ(1) (NPT(J),J=N,NPNS)
      DO 9 I=1,NS
    9 ZOZ(I)=0.
      JJ=NS
C      FOR EACH LINE OF OUTPUT
      DO 10 I=1,NY
C      READ ALL THE POINTS ON THE LINE
      READ (2) (PLT(J),J=1,NS)
C      FOR EACH POINT ON THE LINE
      DO 10 J=1,NS
      JJ=JJ+1
      NN=(NPT(JJ)-1)*NA+PLT(J)
   10 ZOZ(NN)=ZOZ(NN)+1.
      CALL OUTP (ZOZ,NA)
   11 CALL CHNGT
      STOP
C                               ***
C
C                               ***
   12 FORMAT (20HPHOTO CLASSIFICATION)
   13 FORMAT (80A1)
   14 FORMAT (1X,130A1)
      END
```

<>...<><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>

- 23076 WORDS OF MEMORY USED BY THIS COMPILATION

<>...<><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>

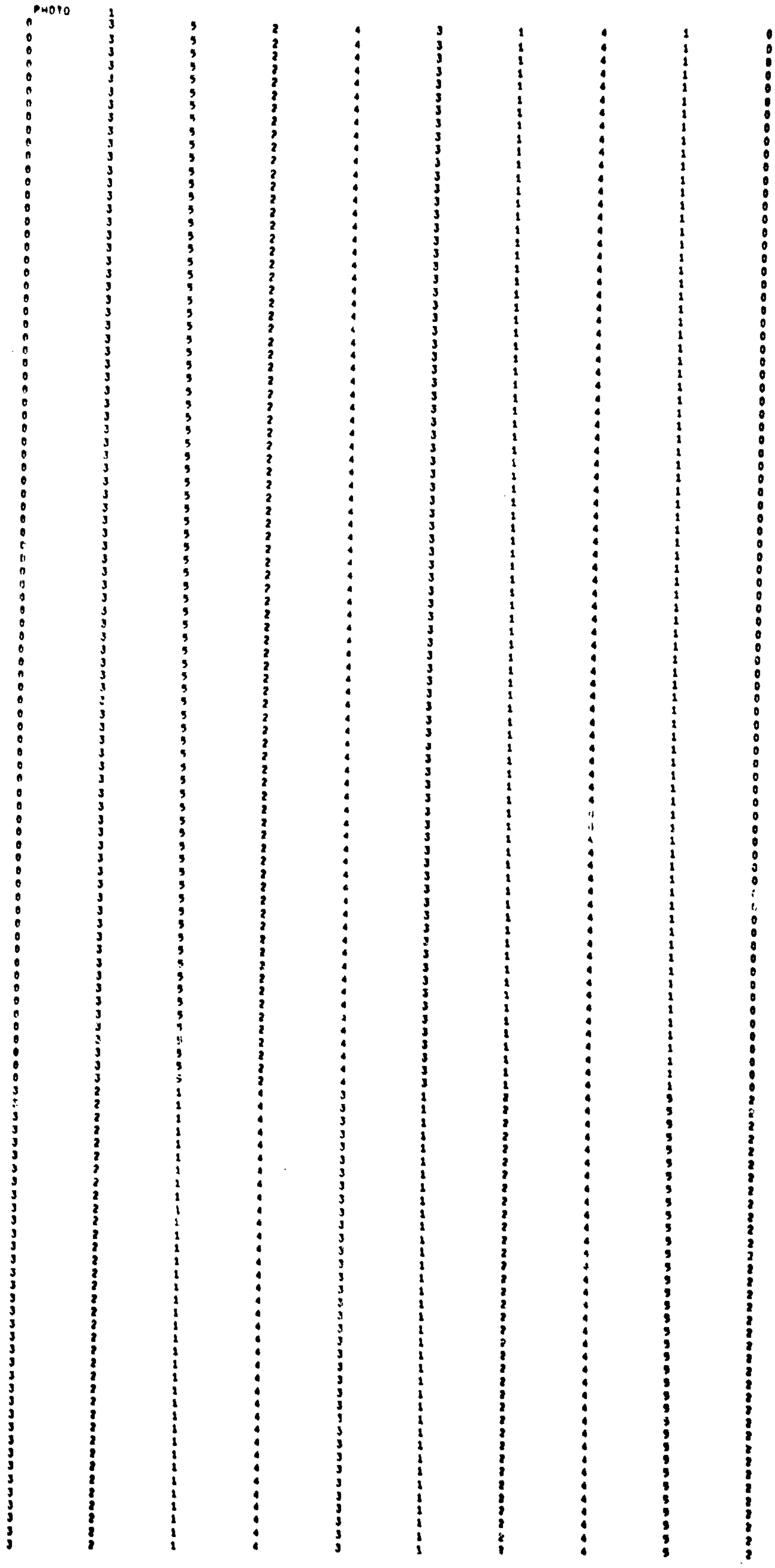
```
17564 03  05-02-69   18.349

      SUBROUTINE OUTP (RESULT,N)
      DIMENSION RESULT(N,N), IDL(20)
      COMMON MP31,FORMT1(12)
      COMMON F
      COMMON FMT1(15),FMT2(3),FMT9(5),FMT11(6),FMT12(7),FMT13(3),FMT14(2
     1),FMT16(2),FMT22(3),FMT56(6),FMT57(4),FMT58(8)
      COMMON L8,L9,L10,L11,L12,L13,L14
      INTEGER F
      DATA IDL/1HT,1HR,1HU,1HE,1H ,1HT,1HR,1HA,1HI,1HN,1HI,1HN,1HG,1H ,1
     1HR,1HE,1HG,1HI,1HO,1HN/
C                                ...
C     PRINT OUT PROBABILITY MATRIX IN PUBLISHABLE FASHION
C                                ...
      II=1
      III=8
      NN=N
    1 CALL HNAME
      K=NN
      IF (K.LT.8) GO TO 9
      LINE=0
      K=8
      NS=25-N
      IF (NS.EQ.0) GO TO 3
      DO 2 I=1,NS
      LINE=LINE+1
    2 WRITE (6,15)
    3 LINE=LINE+7
      WRITE (6,16)
      WRITE (6,FMT12)
      WRITE (6,FMT13) (I,I=II,III)
      IF (LINE.NE.24) GO TO 4
      WRITE (6,FMT14) IDL(1)
    4 DO 6 J=1,N
      LINE=LINE+1
      WRITE (6,FMT57) J,(RESULT(I,J),I=II,III)
      IF (44-LINE.LE.0) GO TO 5
      L=LINE-23
      IF (L.LE.0) GO TO 5
      WRITE (6,FMT14) IDL(L)
    5 LINE=LINE+1
      WRITE (6,15)
      IF (44-LINE.LT.0) GO TO 6
      L=LINE-23
      IF (L.LE.0) GO TO 6
      WRITE (6,FMT14) IDL(L)
    6 CONTINUE
    7 LINE=LINE+1
      L=LINE-23
      IF (L.GT.20) GO TO 8
      WRITE (6,FMT16) IDL(L)
      GO TO 7
    8 NN=NN-K
      II=II+8
      III=III+8
      IF (NN.GT.0) GO TO 1
      RETURN
    9 DO 10 I=1,23
   10 WRITE (6,15)
      WRITE (6,16)
      WRITE (6,FMT12)
      III=II+K-1
      WRITE (6,FMT13) (I,I=II,III)
      WRITE (6,FMT16) ID'(1)
      LINE=24
      'O 12 J=1,N
      _INE=LINE+1
      WRITE (6,FMT57) J,(RESULT(I,J),I=II,III)
      IF (44-LINE.LE.0) GO TO 11
      L=LINE-23
      IF (L.LE.0) GO TO 11
      WRITE (6,FMT14) IDL(L)
   11 LINE=LINE+1
      WRITE (6,15)
      IF (44-LINE.LE.0) GO TO 12
      L=LINE-23
      IF (L.LE.0) GO TO 12
      WRITE (6,FMT14) IDL(L)
   12 CONTINUE
   13 LINE=LINE+1
      L=LINE-23
      IF (L.GT.20) GO TO 14
      WRITE (6,FMT16) IDL(L)
      GO TO 13
   14 RETURN
C                                ...
C
C                                ...
   15 FORMAT (/)
   16 FORMAT (57X,17HCONTINGENCY TABLE)
      END

<><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><
*    22939 WORDS OF MEMORY USED BY THIS COMPILATION
<><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><
```

```
$      FORTRAN NDECK,COMDK                                          SEARCH
       SUBROUTINE SEARCH (KPT,KLAS,KK,SYMBOL,J2,NP,K,LEVEL)
       DIMENSION KPT(2), KLAS(2), SY(36)
       DIMENSION KP(100)
       DIMENSION LIST(2)
       COMMON /ISTT/ IST
       DATA SY/1HA,1HB,1HC,1HD,1HE,1HF,1HG,1HH,1HI,1HJ,1HK,1HL,1HM,1HN,1H
      1O,1HP,1HQ,1HR,1HS,1HT,1HU,1HV,1HW,1HX,1HY,1HZ,1H1,1H2,1H3,1H4,1H5,
      21H6,1H7,1H8,1H9,1H0/
C       K IS THE CATEGORY NUMBER
C       SYMBOL IS THE CATEGORY CODE
C       KK IS THE COMPACTED NTUPLE
C       J2 IS UPPER LIMIT FOR KPT AND KLAS ARRAYS
C       LOOK FOR COMPACTED N-TUPLE KK IN LIST
       KS=0
       KL=J2+1
       KTRY=(KS+KL)/2
     1 CONTINUE
       IF(KK-KPT(KTRY)) 3,4,2
     2 IF ((KL-KS).LE.1) GO TO 5
       KL=KTRY
       KTRY=(KL+KS+1)/2
       GO TO 1
     3 IF (KL-KS.LE.1) GO TO 5
       KS=KTRY
       KTRY=(KL+KS)/2
       GO TO 1
     4 K=KLAS(KTRY)
       SYMBOL=SY(K)
       RETURN
     5 CONTINUE
C       KK CANNOT BE FOUND DO NEAREST NEIGHBOR SEARCH
       I1=IDIST(KK,KPT(1),NP,LEVEL)
       L=1
       KP(1)=1
       DO 7 J=2,J2
       I2=IDIST(KK,KPT(J),NP,LEVEL)
       IF (I1.LT.I2) GO TO 7
       IF (I1.NE.I2) GO TO 6
       IF(L.GT.99) GO TO 7
       L=L+1
       KP(L)=J
       GO TO 7
     6 CONTINUE
       I1=I2
       L=1
       KP(1)=J
     7 CONTINUE
       X=RCH(IST)
       K=X*FLOAT(L-1)+1.5
C       KP IS THE ARRAY CONTAINING THE INDEXES FOR ALL THOSE POINTS IN
C       THE KPT ARRAY WHICH ARE CLOSEST TO KK
C       L IS THE UPPER LIMIT OF KP
C       WE WILL CHOOSE ONE POINT FROM THE KP ARRAY AT RANDOM AND
C       IDENTIFY KK WITH THE CATEGORY OF THE POINT IN KPT ASSOCIATED WITH
C       THE RANDOMLY CHOSEN ONE
       K=KP(K)
     8 K=KLAS(K)
       SYMBOL=SY(K)
       RETURN
       END
```

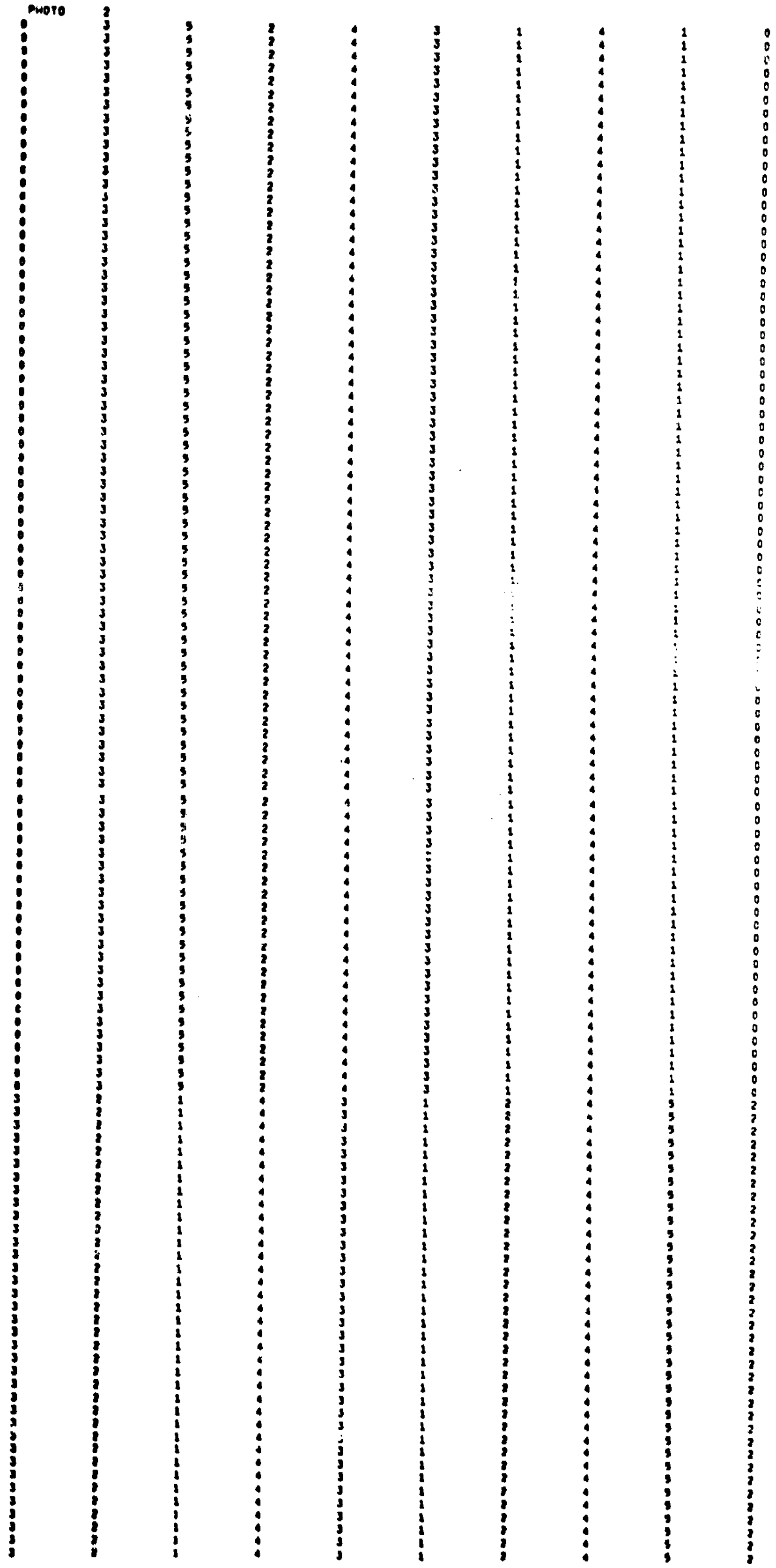


```
$      FORTRAN NDECK,COMDK                                          IDIST
       FUNCTION IDIST(KG,KP,NP,LEVEL)
       IDIST=0
       KK=KG
       LL=KP
       NP1=NP-1
       DO 1 J=1,NP1
       K=KK/LEVEL
       L1=KK-K*LEVEL
       I=LL/LEVEL
       L2=LL-I*LEVEL
       KK=K
       LL=I
     1 IDIST=IDIST+(L1 -L2)*(L1-L2)
       IDIST=IDIST+(K-I)*(K-I)
       RETURN
       END
```

```
SNUMB = 17564, ACTIVITY # = 04, REPORT CODE = 06, RECORD COUNT = 05082

          PHOPTS PATERN DIAGON MLFNMF
          STDPNT PHOUTS
   2        8000         6         2
  20         400
FORMAT FOR TRAINING DATA IS (20A1)


                                         TEST DATA

ORIGINAL DATA

INPUT FORMAT FOR DATA WAS (20I1)                    PHOTO    1                                         PHOTO    2
OUTPUT FORMAT FOR DATA IS (1X,20I1)     0352431410          0352431410          0352431410          0352431410
                                        0352431410          0352431410          0352431410          0352431410
                                        0352431410          0352431410          0352431410          0352431410
                                        0352431410          0352431410          0352431410          0352431410
                                        0352431410          0352431410          0352431410          0352431410
                                        0352431410          0352431410          0352431410          0352431410
                                        0352431410          0352431410          0352431410          0352431410
                                        0352431410          0352431410          0352431410          0352431410
                                        0352431410          0352431410          0352431410          0352431410
                                        0352431410          0352431410          0352431410          0352431410
                                        0352431410          0352431410          0352431410          0352431410
                                        0352431410          0352431410          0352431410          0352431410
                                        0352431410          0352431410          0352431410          0352431410
                                        [illegible]         0352431410          0352431410          0352431410
                                        0352431410          0352431410          0352431410          0352431410
                                        0352431410          0352431410          0352431410          0352431410
                                        0352431410          0352431410          0352431410          0352431410
                                        0352431410          0352431410          0352431410          0352431410
                                        0352431410          0352431410          0352431410          0352431410
                                        0352431410          0352431410          0352431410          0352431410
                                        0352431410          0352431410          0352431410          0352431410
                                        0352431410          0352431410          0352431410          0352431410
                                        [illegible]         0352431410          0352431410          0352431410
                                        0352431410          3214312452          0352431410          3214312452
                                        0352431410          3214312452          0352431410          3214312452
                                        0352431410          3214312452          0352431410          3214312452
                                        0352431410          3214312452          0352431410          3214312452
                                        0352431410          3214312452          0352431410          3214312452
                                        0352431410          3214312452          0352431410          3214312452
                                        0352431410          3214312452          0352431410          3214312452
                                        0352431410          3214312452          0352431410          3214312452
                                        0352431410          3214312452          0352431410          3214312452
                                        0352431410          3214312452          0352431410          3214312452
                                        0352431410          3214312452          0352431410          3214312452
                                        0352431410          3214312452          0352431410          3214312452
                                        0352431410          3214312452          0352431410          3214312452
                                        0352431410          3214312452          0352431410          3214312452
                                        0352431410          3214312452          0352431410          3214312452
                                        0352431410          3214312452          0352431410          3214312452
                                        0352431410          3214312452          0352431410          3214312452
                                        0352431410          3214312452          0352431410          3214312452
                                        0352431410          3214312452          0352431410          3214312452
                                        0352431410          3214312452          0352431410          3214312452
                                        0352431410          3214312452          0352431410          3214312452
                                        0352431410          3214312452          0352431410          3214312452
                                        0352431410          3214312452          0352431410          3214312452
                                        0352431410          3214312452          0352431410          3214312452
                                        0352431410          3214312452          0352431410          3214312452
                                        0352431410          3214312452          0352431410          3214312452
                                        0352431410          3214312452          0352431410          3214312452
                                        0352431410          3214312452          0352431410          3214312452
                                        0352431410          3214312452          0352431410          3214312452
                                        0352431410          3214312452          0352431410          3214312452
                                        0352431410          3214312452          0352431410          3214312452
                                        0352431410          3214312452          0352431410          3214312452
                                        0352431410          3214312452          0352431410          3214312452
                                        0352431410          3214312452          0352431410          3214312452
                                        0352431410          3214312452          0352431410          3214312452
```



PHOTO



PHOTO

TEST DATA

LABLE TABLE

PLOT SYMBOLS VS INPUT SYMBOLS

PLOT SYMBOLS	INPUT SYMBOLS
A	A
B	B

TEST DATA

TRAINING REGIONS

```
ABBABBBABA
ABBABBBABA
ABBABBBABA
ABBABBBABA
ABBABBBABA
ABBABBBABA
ABBABBBABA
ABBABBBABA
ABBABBBABA
ABBABBBABA
ABBABBBABA
ABBABBBABA
ABBABBBABA
ABBABBBABA
ABBABBBABA
ABBABBBABA
ABBABBBABA
ABBABBBABA
ABBABBBABA
ABBABBBABA
ABBABBBABA
ABBABBBABA
ABBABBBABA
ABBABBBABA
ABBABBBABA
ABBABBBABA
ABBABBBABA
ABBABBBABA
ABBABBBABA
ABBABBBABA
ABBABBBABA
ABBABBBABA
ABBABBBABA
ABBABBBABA
ABBABBBABA
ABBABBBABA
ABBABBBABA
ABBABBBABA
ABBABBBABA
ABBABBBABA
ABBABBBABA
ABBABBBABA
ABBABBBABA
ABBABBBABA
ABBABBBABA
ABBABBBABA
ABBABBBABA
ABBABBBABA
ABBABBBABA
ABBABBBABA
ABBABBBABA
ABBABBBABA
ABBABBBABA
ABBABBBABA
ABBABBBABA
ABBABBBABA
```

```
ABBABBBABA
ABBABBBABA
ABBABBBABA
ABBABBBABA
ABBABBBABA
ABBABBBABA
ABBABBBABA
ABBABBBABA
ABBABBBABA
ABBABBBABA
ABBABBBABA
ABBABBBABA
ABBABBBABA
ABBABBBABA
ABBABBBABA
ABBABBBABA
ABBABBBABA
ABBABBBABA
ABBABBBABA
ABBABBBABA
ABBABBBABA
ABBABBBABA
ABBABBBABA
ABBABBBABA
ABBABBBABA
ABBABBBABA
AAAAAAABBB
AAAAAAABBB
AAAAAAABBB
AAAAAAABBB
AAAAAAABBB
AAAAAAABBB
AAAAAAABBB
AAAAAAABBB
AAAAAAABBB
AAAAAAABBB
AAAAAAABBB
AAAAAAABBB
AAAAAAABBB
AAAAAAABBB
AAAAAAABBB
AAAAAAABBB
AAAAAAABBB
AAAAAAABBB
AAAAAAABBB
AAAAAAABBB
AAAAAAABBB
AAAAAAABBB
AAAAAAABBB
AAAAAAABBB
AAAAAAABBB
AAAAAAABBB
AAAAAAABBB
AAAAAAABBB
AAAAAAABBB
AAAAAAABBB
AAAAAAABBB
AAAAAAABBB
AAAAAAABBB
```

TEST DATA

NO. TRAINING REGIONS	NORMALIZED NO.LEVELS	NO. PHOTOS
2	6	2

TEST DATA

THE PROBABILITY MATRIX

TRAINING REGION CLASSIFIED AS

TRUE TRAINING REGION	1	2
1	0.4545E 00	0.5455E 00
2	0.1111E 00	0.8889E 00

TOTAL PROBABILITY

0.6500E 00

N-TUPLES

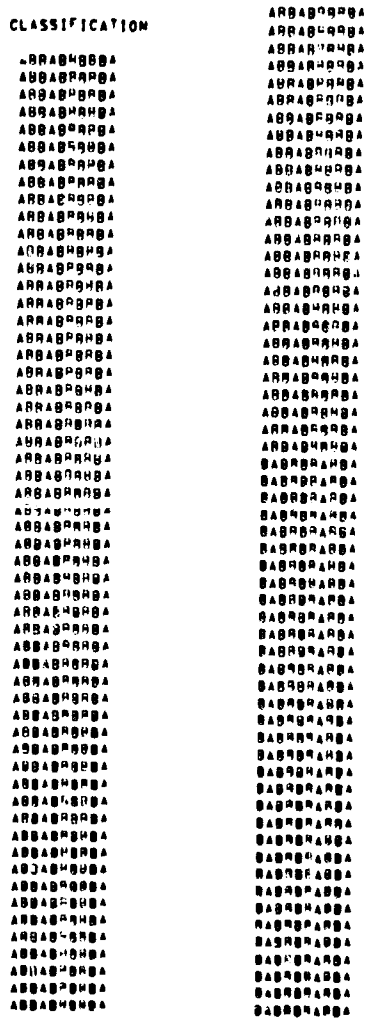
TRAINING REGION

	1	2
35	0	400
28	400	400
21	400	400
14	600	200
7	400	400
0	400	0

35 2 28 2 21 2 14 1 7 2
0 1

TEST DATA

PHOTO CLASSIFICATION



TEST DATA

CONTINGENCY TABLE

TRAINING REGION CLASSIFIED AS

TRUE TRAINING REGION	1	2
1	0.1000E 04	0.1200E 04
2	0.2000E 03	0.1600E 04

SECOND HALF, NUMBER OF POINTS IS 4000

PHOTO 1		PHOTO 2	
1403552510	1403552510	1403552510	1403552510
1403552510	1403552510	1403552510	1403552510
1403552510	1403552510	1403552510	1403552510
1403552510	1403552510	1403552510	1403552510
1403552510	1403552510	1403552510	1403552510
1403552510	1403552510	1403552510	1403552510
1403552510	1403552510	1403552510	1403552510
1403552510	1403552510	1403552510	1403552510
1403552510	1403552510	1403552510	1403552510
1403552510	1403552510	1403552510	1403552510
1403552510	1403552510	1403552510	1403552510
1403552510	1403552510	1403552510	1403552510
1403552510	1403552510	1403552510	1403552510
1403552510	1403552510	1403552510	1403552510
1403552510	1403552510	1403552510	1403552510
1403552510	1403552510	1403552510	1403552510
1403552510	1403552510	1403552510	1403552510
1403552510	1403552510	1403552510	1403552510
1403552510	1403552510	1403552510	1403552510
1403552510	1403552510	1403552510	1403552510
1403552510	1403552510	1403552510	1403552510
1403552510	1403552510	1403552510	1403552510
1403552510	1403552510	1403552510	1403552510
1403552510	5025201230	1403552510	5025201230
1403552510	5025201230	1403552510	5025201230
1403552510	5025201230	1403552510	5025201230
1403552510	5025201230	1403552510	5025201230
1403552510	5025201230	1403552510	5025201230
1403552510	5025201230	1403552510	5025201230
1403552510	5025201230	1403552510	5025201230
1403552510	5025201230	1403552510	5025201230
1403552510	5025201230	1403552510	5025201230
1403552510	5025201230	1403552510	5025201230
1403552510	5025201230	1403552510	5025201230
1403552510	5025201230	1403552510	5025201230
1403552510	5025201230	1403552510	5025201230
1403552510	5025201230	1403552510	5025201230
1403552510	5025201230	1403552510	5025201230
1403552510	5025201230	1403552510	5025201230
1403552510	5025201230	1403552510	5025201230
1403552510	5025201230	1403552510	5025201230
1403552510	5025201230	1403552510	5025201230
1403552510	5025201230	1403552510	5025201230
1403552510	5025201230	1403552510	5025201230
1403552510	5025201230	1403552510	5025201230
1403552510	5025201230	1403552510	5025201230
1403552510	5025201230	1403552510	5025201230
1403552510	5025201230	1403552510	5025201230
1403552510	5025201230	1403552510	5025201230
1403552510	5025201230	1403552510	5025201230
1403552510	5025201230	1403552510	5025201230
1403552510	5025201230	1403552510	5025201230
1403552510	5025201230	1403552510	5025201230
1403552510	5025201230	1403552510	5025201230
1403552510	5025201230	1403552510	5025201230
1403552510	5025201230	1403552510	5025201230
1403552510	5025201230	1403552510	5025201230
1403552510	5025201230	1403552510	5025201230
1403552510	5025201230	1403552510	5025201230

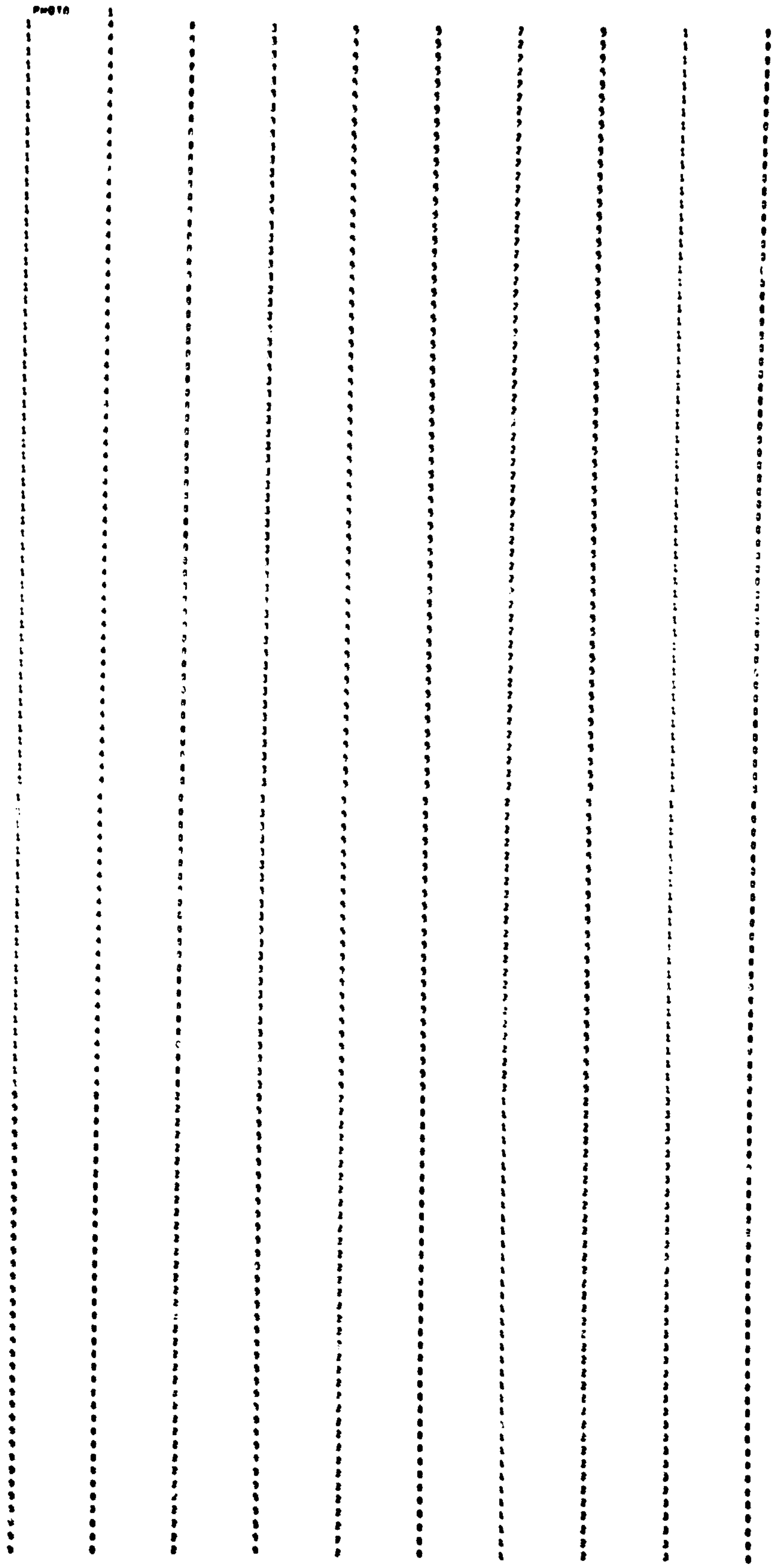


PHOTO	2								
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
1	4	0	3	5	5	2	5	1	0
5	0	2	5	2	0	1	2	3	0
5	0	2	5	2	0	1	2	3	0
5	0	2	5	2	0	1	2	3	0
5	0	2	5	2	0	1	2	3	0
5	0	2	5	2	0	1	2	3	0
5	0	2	5	2	0	1	2	3	0
5	0	2	5	2	0	1	2	3	0
5	0	2	5	2	0	1	2	3	0
5	0	2	5	2	0	1	2	3	0
5	0	2	5	2	0	1	2	3	0
5	0	2	5	2	0	1	2	3	0
5	0	2	5	2	0	1	2	3	0
5	0	2	5	2	0	1	2	3	0
5	0	2	5	2	0	1	2	3	0
5	0	2	5	2	0	1	2	3	0
5	0	2	5	2	0	1	2	3	0
5	0	2	5	2	0	1	2	3	0
5	0	2	5	2	0	1	2	3	0
5	0	2	5	2	0	1	2	3	0
5	0	2	5	2	0	1	2	3	0
5	0	2	5	2	0	1	2	3	0
5	0	2	5	2	0	1	2	3	0
5	0	2	5	2	0	1	2	3	0
5	0	2	5	2	0	1	2	3	0
5	0	2	5	2	0	1	2	3	0
5	0	2	5	2	0	1	2	3	0
5	0	2	5	2	0	1	2	3	0
5	0	2	5	2	0	1	2	3	0
5	0	2	5	2	0	1	2	3	0
5	0	2	5	2	0	1	2	3	0
5	0	2	5	2	0	1	2	3	0
5	0	2	5	2	0	1	2	3	0
5	0	2	5	2	0	1	2	3	0
5	0	2	5	2	0	1	2	3	0
5	0	2	5	2	0	1	2	3	0
5	0	2	5	2	0	1	2	3	0

TEST DATA

LABLE TABLE

PLOT SYMBOLS VS INPUT SYMBOLS

A	A
B	B

TEST DATA

```
ABAABBABAA
ABAABBABAA
ABAABBABAA
ABAABBABAA
ABAABBABAA
ABAABBABAA
ABAABBABAA
ABAABBABAA
ABAABBABAA
ABAABBABAA
ABAABBABAA
ABAABBABAA
ABAABBABAA
ABAABBABAA
ABAABBABAA
ABAABBABAA
ABAABBABAA
ABAABBABAA
ABAABBABAA
ABAABBABAA
ABAABBABAA
ABAABBABAA
ABAABBABAA
ABAABBABAA
ABAABBABAA
ABAABBABAA
AAAAAABBBB
AAAAAABBBB
AAAAAABBBB
AAAAAABBBB
AAAAAABBBB
AAAAAABBBB
AAAAAABBBB
AAAAAABBBB
AAAAAABBBB
AAAAAABBBB
AAAAAABBBB
AAAAAABBBB
AAAAAABBBB
AAAAAABBBB
AAAAAABBBB
AAAAAABBBB
AAAAAABBBB
AAAAAABBBB
AAAAAABBBB
AAAAAABBBB
AAAAAABBBB
AAAAAABBBB
AAAAAABBBB
AAAAAABBBB
AAAAAABBBB
AAAAAABBBB
AAAAAABBBB
AAAAAABBBB
AAAAAABBBB
AAAAAABBBB
AAAAAABBBB
AAAAAABBBB
AAAAAABBBB
```

TRAINING REGIONS

```
ABAABBABAA
ABAABBABAA
ABAABBABAA
ABAABBABAA
ABAABBABAA
ABAABBABAA
ABAABBABAA
ABAABBABAA
ABAABBABAA
ABAABBABAA
ABAABBABAA
ABAABBABAA
ABAABBABAA
ABAABBABAA
ABAABBABAA
ABAABBABAA
ABAABBABAA
ABAABBABAA
ABAABBABAA
ABAABBABAA
ABAABBABAA
ABAABBABAA
ABAABBABAA
ABAABBABAA
ABAABBABAA
ABAABBABAA
ABAABBABAA
ABAABBABAA
ABAABBABAA
ABAABBABAA
ABAABBABAA
ABAABBABAA
ABAABBABAA
ABAABBABAA
ABAABBABAA
ABAABBABAA
ABAABBABAA
ABAABBABAA
ABAABBABAA
ABAABBABAA
ABAABBABAA
ABAABBABAA
ABAABBABAA
ABAABBABAA
ABAABBABAA
ABAABBABAA
ABAABBABAA
ABAABBABAA
ABAABBABAA
ABAABBABAA
ABAABBABAA
ABAABBABAA
ABAABBABAA
ABAABBABAA
ABAABBABAA
ABAABBABAA
```

TEST DATA

PHOTO CLASSIFICATION

	BRARBPAHBA
	RRARBHARBA
	RRAHRHARBA
BBAPBPAHBA	RHAARPARRA
PRAHBPAHBA	RRAHBFARBA
BHARBRARBA	RBARBPAHBA
BBAPBHARBA	BRARBHAPBA
BHAPBPARBA	RRARBPARBA
BRAPBPAPBA	ARARBPAHBA
BHAPBPAPBA	RBARBFAHBA
RBAHBRAPBA	PBARBRARBA
RBARBRAHBA	RRAHBRAPBA
HRARBHARBA	BBARBRARBA
RRARBRAHBA	RBARBPARBA
RRARBRARBA	BRAHBRAHBA
BRAPBHAPBA	BRARBPARBA
RBARBRARBA	RRARBRAPBA
ORARBPARRA	RRARBRAYBA
RBARBPAPBA	RBAPBHAHBA
RBARBPABBA	BRARBHARBA
PHAPBRARBA	RBARBPARBA
PRARBRARBA	RBARBPARBA
R3AHBPARBA	BBARBPARBA
RBARBPAPBA	RBARBPARBA
BBAPBRAPBA	RRARBRARBA
RBARBPAHBA	BBARBRAPBA
PRARBHARBA	BAARAARABA
PRARBPARBA	RAARAABABA
PRARBPARBA	BAARAABABA
RRARBHARRA	RAARAABABA
BBARBRARBA	BAARAABABA
BBAPBRAPRA	BAARAABABA
BBARBRAHBA	BAARAARABA
RBARBRAPBA	BAARAARABA
BRARBHARRA	BAARAABABA
RBARBFAPBA	RAARAARABA
PBARBHAPBA	BAARAABABA
RRAPBHAPBA	RAARAABABA
BRAPBRAHBA	BAABAABABA
BBAPBPARBA	RAARAABABA
RRAPBPABBA	BAARAARABA
BRARBPAPBA	RAARAABABA
BRABBRAPRA	BAARAABABA
RRARBHARBA	BAARAARABA
BBARBRABBA	BAARAARABA
PRARBHAHBA	RAARAABABA
BBAPBRARDA	RAARAABABA
RHARBRAPBA	BAARAARABA
BBARBPARBA	RAARAABABA
HBAPBPAPRA	BAARAAB:DA
RHARBRABBA	BAARAABABA
RRABBPAPBA	BAARAARABA
RBARBRARBA	RAARAARABA
RBARBRAPBA	RAARAABABA
BBARBPARRA	RAARAABARA
RBARBRAPBA	BAARAARABA
RBAPBPARBA	RAABAABABA
BRAPBRARBA	BAARAABABA
BRABBRABBA	RAARAABABA

TEST DATA

CONTINGENCY TABLE

TRAINING REGION CLASSIFIED AS

TRUE TRAINING REGION	1	2
1	0.1400E 04	0.1000E 04
2	0.4000E 03	0.1200E 04

Unclassified
Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)* Center for Research, Inc. The University of Kansas	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

3. REPORT TITLE

The Bayesian Approach to Identification of a Remotely Sensed Environment

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*

5. AUTHOR(S) *(First name, middle initial, last name)*

Robert M. Haralick

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
July 1969	68	

8a. CONTRACT OR GRANT NO. DAAK02-68-C-0089 b. PROJECT NO. ARPA Order No. 1079 c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) 133-9 9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)* Supported

10. DISTRIBUTION STATEMENT

No Limitation

11. SUPPLEMENTARY NOTES Funded by DOD Project Themis under ARPA Order No. 1079	12. SPONSORING MILITARY ACTIVITY Monitoring: U.S. Army Engineer Topographic Labs Geographic Information Systems Branch Geographic System Division Ft. Belvoir, Virginia

13. ABSTRACT

The first part of this paper provides a brief tutorial introduction of the Bayesian Approach to identification of a remotely sensed environment. The second part describes the input data deck setup for the Fortran IV program which has been written to implement this approach. The third part describes file usage and subroutine organization. The fourth part provides a listing of the program with a simple sample data set.

DD FORM 1 NOV 65 1473 REPLACES DD FORM 1473, 1 JAN 64, WHICH IS OBSOLETE FOR ARMY USE.

Unclassified
Security Classification

CRES LABORATORIES

Chemical Engineering Low Temperature Laboratory

Remote Sensing Laboratory

Electronics Research Laboratory

Chemical Engineering Heat Transfer Laboratory

Nuclear Engineering Laboratory

Environmental Health Engineering Laboratory

Digital Computer Technology Laboratory

Water Resources Institute